

SOFTWARE STARTUP PRODUCT MANAGEMENT

A Multi-Method Study Leading to the Pragmatic Framework for Product Managers (PFPM) and VECTR, a Startup-Appropriate Requirements Prioritization Method.

Frédéric Pattyn

Supervisor: Prof. Dr. Frederik Gailly & Prof. Dr. Yannick Dillen

A dissertation submitted to Ghent University in partial fulfilment of the requirements for the degree of Doctor of Business Economics

Academic year: 2025 – 2026

Copyright © 2026 by Frédéric Pattyn

All rights are reserved. No part of this publication may be reproduced or transmitted in any form or by any means electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from the author.

DOCTORAL ADVISORY COMMITTEE

Prof. Dr. Geert Poels (Ghent University, Belgium)
Prof. Dr. Frederik Gailly (Supervisor, Ghent University, Belgium)
Prof. Dr. Yannick Dillen (Supervisor, Ghent University, Belgium)
Prof. Dr. Petra Andries (Ghent University, Belgium)

EXAMINATION BOARD

Prof. dr. Broos Maenhout (Chair of the Doctoral Examination Committee, Ghent University, Belgium)
Prof. Dr. Geert Poels (Secretary of the Doctoral Examination Committee, Ghent University, Belgium)
Prof. Dr. Frederik Gailly (Supervisor, Ghent University, Belgium)
Prof. Dr. Yannick Dillen (Supervisor, Ghent University, Belgium)
Prof. Dr. Petra Andries (Ghent University, Belgium)
Prof. Dr. Pekka Abrahamsson (Tampere University, Finland)
Prof. Dr. Xiaofeng Wang (Free University of Bozen-Bolzano, Italy)

LIST OF PUBLICATIONS AND CONFERENCE PRESENTATIONS BASED ON THIS DOCTORAL RESEARCH

Chapter 2 based on:

Pattyn, F. (2024). The Critical Role of Product Managers and Their Responsibilities in Software Startups: A Systematic Literature Review. *American Journal of Engineering and Technology Management*, 9(4). **(published journal)**
and

Pattyn, F. (2024). Innovating the Future: Unveiling the Initial Iteration of the Pragmatic Framework for Product Managers. *American Journal of Engineering and Technology Management*, 9(2). **(published journal)**

Chapter 3 based on:

Measuring Role Concentration in Product Management: An Attention-Based Analysis Using the Herfindahl-Hirschman Index. **(submitted to Information Systems Technology)**

Chapter 4 based on:

Validate Before You Build: Exploring Pre-MVP Evidence Levels – Not Quantity – and Startup Performance in Early-Stage Software Ventures (Working paper)

Chapter 5 based on:

What Really Matters in Requirements Prioritization: Evidence from 171 Practitioners in Software Startup and Mature Company Contexts. (Working paper).

Chapter 6 based on:

Pattyn, F., & Goetz, P. (2025). Time to Value (TtV) as a Missing Requirements Prioritization Criterion for Startup Survival: Insights from Founders vs. Systematic Literature Review. *INTERNATIONAL JOURNAL OF SUSTAINABILITY MANAGEMENT AND INFORMATION TECHNOLOGIES*, 11(1), 45-66. **(published journal)**
and

Pattyn, F., & Rafiq, U. (2024, November). Smart Scaling for Software Startups Through Financial Requirements Prioritization Criteria. In *International Conference on Software Business* (pp. 454-460). Cham: Springer Nature Switzerland. **(published conference)**

Chapter 7 based on:

Pattyn, F., Wang, X., Poels, G., Dillen, Y., & Goetz, P. (2026). VECTR: A Lightweight Requirements Prioritization Method for Software Startups Grounded in ROI, Time-to-Value, and Confidence. *Computers*, 15(2), 117. <https://doi.org/10.3390/computers15020117>
(published journal)

ACKNOWLEDGEMENTS

This dissertation marks the culmination of a journey that has shaped me far beyond the boundaries of research. It has taught me humility in the face of so many brilliant minds, perseverance in moments of doubt, and the immense power of teamwork.

First and foremost, I am deeply grateful to my supervisors, Frederic Gailly and Yannick Dillen, for taking a chance on me and my topic, and to Geert Poels for his steady support. Their trust made this path possible. I also thank the members of my commission and jury for their thorough evaluation and invaluable feedback, which sharpened both this dissertation and my own thinking.

To the ICSOB conference, our annual gathering where friendships for life are formed, and to my closest collaborators who share that spirit: Usman, with his heart of gold and writing talent I will always admire—you have been more than a colleague; you have been a companion on this road. Xiao, whose warmth and ability to bring people together outshine even her academic brilliance, I am grateful to have learned from your team what collective achievement truly means. Slinger, thank you for reminding me that speed is never a substitute for quality. Pekka, your drive and vision are unmatched. You inspire us not just to publish, but to change the world. I am honored that you invested your time and belief in me.

To those who have supported me personally: Yasmine, my wife, my greatest cheerleader, whose unwavering trust in me carried me through every rejection and every doubt. You are the reason I dared to believe in myself. Robin, my son—you are my deepest motivation, the reason I show up and push forward. I hope you will one day look back with pride and know that hard work and perseverance can turn dreams into reality. To my family—my mother and Luc—thank you for pushing me, sometimes fiercely, but always with love. I dedicate this milestone in part to my mother, who was never given the chance to pursue her own academic ambitions.

I am grateful to Hans-Bernd, for bridging the academic and professional worlds and opening doors that shaped my early career; to Peter Goetz, whose respect, support, and relentless drive have deeply inspired me and carried me forward; and to The House of HR and NOWJOBS, for reminding me of my strength as a product manager and giving me the confidence to pursue this PhD. Finally, my thanks go to Beatrice De Keyzer and Lannoo, for believing in my work and giving me the opportunity to turn it into a book that I hope will inspire future founders.

This dissertation is the product of many minds, many hearts, and many helping hands. To each of you who shaped my journey: thank you. I carry your influence with me, and this achievement belongs as much to you as it does to me.

*Ghent, June 2026
Frédéric Pattyn*

NEDERLANDSTALIGE SAMENVATTING

Dit doctoraat onderzoekt softwarestartups, deze opereren in een omgeving van hoge onzekerheid en schaarse middelen. In de vroege fasen nemen oprichters, vaak onbewust, de rol van eerste productmanager op zich. Deze rol omvat meer dan honderd uiteenlopende activiteiten. Doordat oprichters zich meestal niet bewust zijn van de breedte en complexiteit van deze rol, is er weinig houvast over welke activiteiten werkelijk prioriteit verdienen tijdens deze fase. Dit gebrek aan focus vergroot het risico op inefficiënt gebruik van middelen en het nemen van beslissingen die de overlevingskansen van de potentieel startup ondermijnen.

Bestaand onderzoek naar productmanagement richt zich voornamelijk op mature organisaties, waardoor systematische kennis over productmanagement in startups grotendeels ontbreekt. Hoewel verschillende studies geïsoleerde aspecten hebben onderzocht, ontbreekt een breed inzicht in welke activiteiten bepalend kunnen zijn voor succes in de specifieke context van startups. Er bestaan geen duidelijke richtlijnen die founders en product managers kunnen helpen hun schaarse middelen, zowel tijd als kapitaal, te richten op de activiteiten en projecten met potentieel de grootste positieve impact. Daarmee ontstaat een kenniskloof die niet alleen theoretisch maar ook praktisch van groot belang is, aangezien verkeerde prioriteiten de kans op falen aanzienlijk vergroten.

Dit doctoraat beoogt deze leemte te vullen door te onderzoeken welke productmanagement-activiteiten de meeste waarde creëren in softwarestartups en hoe deze verschillen van activiteiten in mature bedrijven. Het doel is niet enkel academisch, maar ook praktisch: founders en productmanagers beter uitrusten om evidence-based beslissingen te nemen en de kans op duurzame groei te vergroten. Om dit te realiseren zijn verschillende onderzoeksmethodes over verschillende studies toegepast zoals onder andere een grootschalige survey onder 171 productmanagers in zowel startups als gevestigde ondernemingen, als meer dan zeventien expertinterviews om de inzichten kwalitatief te verdiepen. Daarnaast worden nieuwe methodieken ontwikkeld en getoetst om startups te ondersteunen bij het prioriteren en valideren van hun beslissingen.

Dit proefschrift levert zes onderling verbonden wetenschappelijke bijdragen. Het Pragmatic Framework for Product Managers (PFPM) biedt een systematische mapping van alle productmanagementactiviteiten, verfijnd voor de context van softwarestartups. De Herfindahl–Hirschman Index (HHI) wordt geïntroduceerd als kwantitatieve maat voor rolconcentratie bij productmanagers, toegepast op 126 practitioners in zowel startup- als mature-bedrijfscontexten. De Hiërarchie van Validatie introduceert een bewijsniveau-model dat founders helpt de sterkte van marktevidentie te beoordelen voordat ze zich vastleggen op ontwikkeling. Twee complementaire empirische studies brengen in kaart welke requirements-prioriteringscriteria het meest relevant zijn in startup- versus mature-contexten, en identificeren Time-to-Value (TtV) als een tot dusver onerkend prioriteringscriterium dat cruciaal is voor de overleving van startups. Tot slot integreert VECTR ROI, TtV en Confidence in één lichtgewicht prioriteringskader, specifiek ontworpen voor softwarestartups.

Door deze inzichten en methodieken samen te brengen, levert dit doctoraat een belangrijke bijdrage aan de literatuur over productmanagement in startups en ontwikkelt het concrete instrumenten die de praktijk direct kunnen ondersteunen. Het onderzoek laat zien dat betere rangschikking en validatie niet alleen academisch relevant zijn, maar ook essentieel om startups te helpen hun beperkte middelen effectiever in te zetten, risico's te beperken en hun kans op duurzaam succes te vergroten.

Het proefschrift is een op papier gebaseerde thesis bestaande uit 8 hoofdstukken. Voor 3 hoofdstukken (hoofdstukken 2, 5 en 7) zijn reeds studies gepubliceerd in internationale peer-reviewed journals of conferenties, terwijl voor de andere 3 (hoofdstukken 3, 4 en 6) werkdocumenten zijn die momenteel worden verfijnd voordat ze (opnieuw) worden ingediend.

SUMMARY IN ENGLISH

This PhD research examines software startups, which operate in environments characterized by high uncertainty and scarce resources. In the early stages, founders often unknowingly assume the role of the first product manager. This role involves more than one hundred diverse activities across domains ranging from Executive Leadership, Product Strategy & Planning, Engineering & Development, and Marketing and Sales, to Customer Support & Success. Since founders are rarely aware of the breadth and complexity of this role, there is little guidance on which activities should be prioritized during this phase. This lack of focus increases the risk of inefficient use of resources and of decisions that may undermine the survival of the startup.

Existing research on product management has primarily concentrated on mature organizations, leaving systematic knowledge about product management in startups largely underexplored. Although several studies have investigated isolated aspects, a comprehensive understanding of which activities may be most critical for success in the specific context of startups is lacking. No clear guidelines exist to help founders and product managers allocate their scarce resources—both time and capital—to the activities and projects with the greatest potential positive impact. This gap is of both theoretical and practical importance, as misplaced priorities substantially increase the likelihood of failure.

This dissertation aims to address this gap by investigating which product management activities create the most value in software startups and how they differ from those in mature companies. The objective is not only academic but also practical: to better equip founders and product managers to make evidence-based decisions and increase their chances of sustainable growth. To achieve this, multiple research methods were applied across several studies, including a large-scale survey of 171 product managers in both startups and established firms, as well as more than seventeen expert interviews to enrich the qualitative insights. In addition, new methods were developed and validated to support startups in prioritizing and validating their decisions.

This dissertation makes six interconnected scientific contributions. The Pragmatic Framework for Product Managers (PFPM) provides a systematic mapping of all product management activities, refined for the specific context of software startups. The Herfindahl–Hirschman Index (HHI) is introduced as a quantitative measure of product management role concentration, applied across 126 practitioners in both startup and mature company contexts. The Hierarchy of Validation introduces an evidence-level model that helps founders evaluate the strength of market evidence before committing to development. Two complementary empirical studies establish which requirements prioritization criteria matter most in startup versus mature contexts, and identify Time-to-Value (TtV) as a previously unrecognized prioritization criterion critical to startup survival. Finally, VECTR integrates ROI, TtV, and Confidence into a single lightweight prioritization framework specifically designed for software startups.

By bringing together these insights and methods, this dissertation makes an important contribution to the literature on product management in startups while also developing concrete instruments that can directly support practice. The research demonstrates that improved prioritization and validation are not only academically relevant but also essential to help startups deploy their limited resources more effectively, mitigate risks, and increase their chances of long-term success.

It is a paper-based thesis consisting of eight chapters. For three chapters (Chapters 2, 5, and 7), studies have already been published in international peer-reviewed journals or conferences, while for three others (Chapters 3, 4, and 6), working papers are currently being refined before (re)submission.

TABLE OF CONTENT

1. Introduction	1
1.1. Research context	2
1.2. Product Management.....	2
1.3. Product Management in Software Startups	3
1.4. Knowledge Gap & Problem Statement	4
1.5. Research Questions	5
1.6. Research Methodology.....	5
1.7. Threats to Validity.....	9
1.8. Dissertation Structure	9
2. Defining the Product Manager Role: A Systematic Literature Review and Organizing Framework of Product Manager Activities	12
2.1. Introduction	13
2.2. Product Management as a Fragmented Activity Domain	14
2.3. Methodology	15
2.4. Findings: The Landscape of Product Management Activities	19
2.5. An Organizing Framework for Product Manager Activities.....	26
2.6. Discussion	31
2.7. Conclusion.....	33
3. Measuring Role Concentration in Product Manager Role: Using the Herfindahl-Hirschman Index: Comparison of Startups and Mature Companies.....	34
3.1. Introduction	35
3.2. Background	37
3.3. Methodology	40
3.4. Results.....	44
3.5. Discussion	48
3.6. Limitations	50
3.7. Conclusion.....	51
4. Validate Before You Build: Exploring Pre-MVP Evidence Levels – Not Quantity – and Startup Performance in Early-Stage Software Ventures	53
4.1. Introduction	54
4.2. Related work	55
4.3. Study design	56
4.4. Results	58
4.5. Discussion	72
4.6. Conclusion.....	75
5. What Really Matters in Requirements Prioritization: Evidence from 171 Practitioners in Software Startup and Mature Company Contexts	77
5.1. Introduction	78
5.2. Related Work.....	79
5.3. Time is Money	81
5.4. Research Methodology.....	83
5.5. Results	86
Demographic data from survey	86
5.6. Discussion	92
5.7. Revenue and financial impact	92
5.8. Threats to Validity.....	94
5.9. Conclusions	95

5.10. Summary	95
5.11. Research Agenda.....	95
6. Financial and Temporal Decision Criteria in Software Startup Prioritization: Introducing Time to Value (TtV) and Reassessing ROI through Founders' Insights vs. Systematic Literature Review	97
6.1. Introduction	98
6.2. Related Work.....	99
6.3. Research Methodology.....	102
6.4. Results	109
6.5. Discussion	123
6.6. Completeness	125
6.7. Threats to Validity.....	125
6.8. Conclusions	126
6.9. Research Agenda.....	127
7. VECTR: A Lightweight Requirements Prioritization Method for Software Startups Grounded in ROI, Time-to-Value, and Confidence	128
7.1. Introduction	129
7.2. Related Work.....	131
7.3. Research Methodology.....	131
7.4. Software Startup Requirements Prioritization Method: VECTR.....	134
7.5. Findings.....	137
7.6. Discussion	143
7.7. Threats to Validity.....	144
7.8. Conclusions	145
8. Conclusion	146
8.1. Research Results and Scientific Contributions	147
8.2. Practical contributions.....	151
8.3. Research limitations	152
8.4. Future research	152
8.5. Closing reflection	153

LIST OF FIGURES

Figure 1: Storyline of the studies used in the dissertation	11
Figure 2: Search, screening and acceptance process	18
Figure 3: Number of validation events	59
Figure 4: Basic Validation Canvas – Three Stages drawn from literature	61
Figure 5: Updated Validation Canvas (3x2) – Three Stages, Six Evidence Levels	66
Figure 6: Tier 1 – Top Performers	66
Figure 7: Tier 2 – Mid-high Performers	67
Figure 8: Tier 3 and Out-of-Business	68
Figure 9: The Hierarchy of Validation as a bidirectional process model.	72
Figure 10: Sources of survey respondents	85
Figure 11: Hybrid coding process.....	86
Figure 12: Counts and proportional selections for Q1 (“Most Important” RPC Category) ...	87
Figure 13: Counts and Proportional Selections for Coded Q2 (“Ideal Process”) Categories.	87
Figure 14: Requirements Prioritization Criteria: Mature companies vs Startups	90
Figure 15: P-values for RPCs and Contextual Factors	91
Figure 13-Figure 16: Overlapping of criteria between startups and mature companies	91
Figure 17: Overview of review protocol.....	105
Figure 18: Protocol result search strategy.....	107
Figure 19: Number of papers by year of publication.....	109
Figure 20: Use of financially related requirements prioritization criteria versus ranking....	122
Figure 21: Conceptual view of the VECTR framework. Color zones indicate strategic priority: green = high confidence + fast TtV (highest priority); red = low confidence + slow TtV (lowest priority).	135
Figure 22: Visual encoding in VECTR: confidence on the x-axis, Time-to-Value on the y- axis, and ROI represented by bubble size.....	136
Figure 23: Example VECTR map. Larger bubbles in the upper-right (high ROI + high confidence + fast TtV) represent highest-priority requirements.....	136
Figure 24: Dynamics of confidence within VECTR. Example scenario: Requirement [X] gains confidence through validation and moves from red to green ([X → X’]); requirement [Y] loses confidence and shifts from orange to red ([Y → Y’]).	137
Figure 25: Thematic structure of findings, linking current practices (RQ1) and practitioner evaluation of VECTR (RQ2).	138

LIST OF TABLES

Table 1: Methodological positioning of each study	9
Table 2: Inclusion criteria	16
Table 3: Exclusion criteria	16
Table 4: Quality assessment criteria	17
Table 5: Distribution of identified activities across publications	19
Table 6: ISPMA versus the Pragmatic Framework for Product Managers (PFPM) domains	27
Table 7: Executive leadership domain: categories and activities	27
Table 8: Product strategy & planning domain: categories and activities.....	28
Table 9: Engineering & development domain: categories and activities.....	29
Table 10: Sales domain: categories and activities	29
Table 11: Marketing domain: categories and activities	30
Table 12: Customer support & success domain: categories and activities	30
Table 13: Herfindahl–Hirschman Index (HHI) thresholds	39
Table 14: Sample Characteristics.....	41
Table 15: Ratio-scale properties of the HHI score as applied in this study	43
Table 16: Time Allocation by Domain and Context.....	44
Table 17: Distribution of HHI Scores	45
Table 18: Classification by Concentration Category	46
Table 19: Summary of Robustness Analysis	47
Table 20: Startups profiles	57
Table 21: Startups and interviews	58
Table 22: Coding results for Validation Canvas, Validation activities coded per startup	62
Table 23: Pre-test (PT) survey answers	84
Table 24: Pilot (P) survey answers	84
Table 25: Demographic Profile of Respondents (n=171)	86
Table 26: Context scores for RPC categories in Q1 and Q2.....	88
Table 27: Logistic regression p-values and context score for RPC category selections (Q1)	88
Table 28: Requirements prioritization methods (alphabetical order)	99
Table 29: Research questions.....	103
Table 30: Inclusion criteria	105
Table 31: Exclusion criteria	106
Table 32: Quality assessment criteria	106
Table 33: Startups and interviews.....	108
Table 34: Overview of distribution of identified requirements prioritization criteria (top 10)	109
Table 35: Selected studies and their RP criteria	110
Table 36: Overview of the RPC across categories.....	112
Table 37: Studies considering a software startup context.....	114
Table 38: Requirements prioritization criteria in a startup context	115
Table 39: Overview of RPC across categories in a software startup context	117
Table 40: Overview of used RPC by software startup practitioners vs literature.....	118
Table 41: Overview of used RPC across categories by software startups vs scholarly investigations	120
Table 42: Startups & performance	122
Table 43: Financial RPC	123

Table 44: Overview of studied cases, interviewees	132
Table 45: Themes of current prioritization practices (RQ1) with representative quotes.....	139
Table 46: Practitioner evaluation of VECTR (RQ2): strengths and illustrative quotes.....	141

ABBREVIATIONS

AAOM	Agile Agent-Oriented Modelling
ACM	Association for Computing Machinery
AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
AIS	Association for Information Systems
AmbiTRUS	Ambiguity Tracking and Resolution for User Stories
API	Application Programming Interface
ASD	Agile Software Development
BDD	Behavior-Driven Development
BOW	Bag-of-Words
BPMN	Business Process Modeling Notation
CAPEX	Capital Expenditure
CASE	Computer-Aided Software Engineering
CBR	Case-based Reasoning
CC	Coordinating Conjunction
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
EBNF	The Extended Baccus-Naur Form
FDD	Feature-Driven Development
INCOSE	International Council on System Engineering
INVEST	Independent, Negotiable, Valuable, Estimable, Small, and Testable
IS	Information Systems
IT	Information Technology
ITU	Intention to Use
LEL	Language Extended Lexicon
LLM	Large Language Model
MEM	Method Evaluation Model
ML	Machine Learning
N/A	Not Available
NL	Natural Language
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
OODARE Engineering	Object-oriented Analysis and Design Approach for the Requirements Engineering
OPEX	Operating Expenses
OWL	The Web Ontology Language
PICOC	Population, Intervention, Comparison, Outcomes, and Context
PO	Product Owner
POS	Part-of-Speech
PU	Perceived Useful
PUOE	Perceived Ease of Use
QUS	Quality User Story
RE	Requirements Engineering
ROI	Return on Investment
RQ	Research Question
RUP	Rational Unified Process
SaFE	Scaled Agile Framework
SE	Software Engineering
SC	Specific Criteria
SMART	Specific, Measurable, Acceptable, Realistic, Time-bound

SPCM	Software Product Certification Model
SRS	Software Requirements Specification
TAM	Technology Acceptance Model
TF-IDF	Term Frequency-Inverse Document Frequency
UML	Unified Modeling Language
US	User Story
VC	Verification Criteria
XP	Extreme Programming

1. Introduction

1.1. Research context

Software startups are widely recognized as engines of innovation, economic growth, and technological disruption. They create new markets, challenge incumbents, and deliver solutions to emerging needs at a pace that established organizations often cannot match. Yet, despite their potential, most startups fail within the first few years of operation. Studies consistently highlight that resource scarcity, market uncertainty, and execution challenges are key drivers of this high mortality rate. For researchers and practitioners alike, this raises an important question: how can startups increase their probability of survival and long-term success?

Central to this question is the role of product-related decision-making. Startups typically succeed or fail on the basis of their product—whether it solves a real customer problem, whether it reaches the market at the right time, and whether limited resources are allocated effectively. Unlike established firms, which often rely on stable processes, specialized departments, and historical data, startups must make these decisions under severe uncertainty and time pressure. The wrong call on which feature to prioritize, which customer segment to target, or how to allocate scarce development capacity can easily exhaust the startup’s runway. This is where product management becomes essential. At its core, product management is about identifying customer needs, aligning them with organizational strategy, and translating them into product decisions that create value. In larger organizations, product management has become a professionalized discipline supported by established frameworks and roles. In startups, however, the situation is quite different. Founders themselves often act as the first product managers, usually without realizing it, and must make decisions that span across vision, strategy, development, marketing, sales, and customer support simultaneously. The breadth of these responsibilities, combined with limited resources, makes prioritization both unavoidable and extremely difficult.

While much of the existing research and practice around product management has been shaped by the needs of mature organizations, systematic knowledge about how which product management activities and criteria in startups actually matter remains limited. This creates a gap between theory and practice: founders are left without clear guidance on which activities matter most, which decision criteria they should emphasize, and how to structure their product management efforts in ways that maximize their survival chances. The result is often intuition-driven, ad hoc decision-making that increases the risk of wasted resources and premature failure.

This dissertation is situated precisely at this intersection. It explores the role of product management in software startups, investigates which activities and prioritization criteria truly matter in the early stages, and develops tools and methods that enable startups to make better product decisions under uncertainty. In doing so, it aims to contribute both to academic understanding of product management in high-uncertainty contexts and to the practical needs of founders navigating the journey from idea to sustainable venture.

1.2. Product Management

Product management has emerged over the past decades as a pivotal discipline within technology-driven organizations [1, 2]. At its essence, product management is concerned with defining what should be built, why it should be built, and how it can deliver value to both customers and the business [3, 4]. It acts as the connective tissue between strategy, design, engineering, marketing, and sales [5]. By aligning market insights with organizational capabilities, product management helps ensure that resources are invested in the right initiatives and that products meet both user needs and strategic objectives [6].

The responsibilities of product management are broad and often fluid [7]. They include shaping the product vision, translating customer insights into requirements, prioritizing features,

coordinating development teams, and monitoring product performance once launched [8]. Unlike purely technical roles, product management is inherently cross-functional and interdisciplinary [9]. It requires a blend of market awareness, strategic thinking, and operational execution [10]. In many organizations, product managers serve as the “voice of the customer,” ensuring that what is developed addresses genuine problems, while simultaneously balancing technical feasibility and business viability [11].

The importance of product management has grown in parallel with the increasing complexity of digital products [12, 13]. As software and digital platforms have become central to industries ranging from finance and healthcare to education and mobility, the need for structured decision-making around product development has intensified. Organizations that excel at product management often outperform competitors by bringing relevant innovations to market faster, capturing customer loyalty, and allocating resources more efficiently [14, 15]. In contrast, organizations that neglect product management risk building products that fail to resonate with users, waste development effort, or lack strategic coherence [10].

Academically, product management is still a relatively young field [7]. Much of the early literature is descriptive, focusing on roles and responsibilities, while more recent studies have examined its strategic dimensions, such as portfolio management, innovation processes, and agile product development [16]. A consistent theme across this body of work is that product management reduces risk in uncertain environments by creating mechanisms for evidence-based decision-making [17]. Techniques such as market validation, customer feedback loops, and structured prioritization frameworks help organizations allocate resources to initiatives that have the greatest chance of success [11].

However, the majority of existing frameworks and models have been developed with mature organizations in mind [18, 19]. These contexts are characterized by relatively stable markets, established customer bases, and formalized processes [20]. Product management in such organizations often revolves around optimizing portfolios, balancing innovation with sustaining products, and coordinating across large teams [21]. While valuable, this perspective only partially reflects the realities of organizations operating under conditions of extreme uncertainty [22].

This distinction is critical for startups [23]. Unlike their mature counterparts, startups lack historical data, established market positions, and process maturity [24]. Their product management challenges are not about balancing portfolios but about survival: identifying which ideas merit scarce investment, which customer feedback to act upon, and which features or experiments are most likely to generate traction [25]. As the next section will elaborate, this makes the startup context not only unique but also underexplored in both research and practice [26]. Understanding how product management functions under such conditions, and what makes it effective, is therefore a pressing need for both scholars and practitioners.

1.3. Product Management in Software Startups

While product management has become increasingly formalized in larger organizations, the situation in software startups is fundamentally different [27]. Startups emerge in contexts defined by scarcity and uncertainty: scarce resources, time, and experience on the one hand, and uncertain markets, technologies, and business models on the other. In such environments, product-related decisions often determine not only the direction of the company but also its very survival [28].

Unlike mature firms that employ specialized product managers, startups frequently rely on their founders to take on the product management role [29]. This often occurs unconsciously: founders are driven by a vision or idea and naturally assume responsibility for setting direction, making prioritization decisions, and interacting with customers. In effect, they become the “first product managers,” but without the explicit recognition of the role or the support of

established processes and frameworks. Their decisions span a wide range of domains—strategy, design, engineering, marketing, sales, and customer support—often simultaneously. This breadth of responsibility creates both opportunity and risk.

Research on product management in startups is still limited compared to the wealth of studies in mature organizations [26]. The literature that does exist highlights several recurring challenges [30]. First, startups face extreme prioritization pressures: with limited time and funding, they must carefully decide which features to develop, which customer segments to pursue, and which markets to test [31]. Second, startups must operate with minimal data [32]. Unlike established companies, they cannot rely on historical analytics or large-scale market studies, and must instead generate evidence through rapid experimentation and customer feedback [33]. Third, startups operate under compressed timelines [34]. Investors, competitors, and customers expect quick results, forcing teams to make high-stakes decisions with incomplete information [35].

These challenges suggest that product management in startups cannot simply replicate the models used in larger firms [31]. Instead, it requires approaches tailored to high-uncertainty environments [24].

1.4. Knowledge Gap & Problem Statement

Although product management is widely acknowledged as critical to the success of technology-driven organizations, the majority of research and practice in this field has been shaped by the needs of mature companies. These organizations typically operate in relatively stable markets, have established product portfolios, and employ specialized product managers supported by formalized processes. As a result, much of the literature on product management emphasizes portfolio optimization, scaling frameworks, and cross-functional coordination in large organizations, assumptions that rarely hold in early-stage startup environments.

Four critical knowledge gaps motivate this dissertation. First, while we know that product management involves a wide range of activities, there is little clarity on which activities are most critical in a startup context — and how these differ from mature organizational practices. Second, existing requirements prioritization research focuses predominantly on technique-oriented solutions for mature, stable environments, offering limited guidance on which decision criteria startup practitioners should prioritize under survival constraints. Third, temporal and financial prioritization dimensions — such as the speed at which value can be realized from a feature decision — remain largely absent from the academic literature despite their practical importance to startups operating under runway pressure. Fourth, while various frameworks exist for mature organizations, startups lack practical, empirically validated tools that reflect their unique constraints.

The core problem this dissertation addresses is therefore:

“Startups lack systematic, evidence-based knowledge and practical tools to guide effective product management activities and prioritization decisions under conditions of high uncertainty and resource scarcity.”

1.5. Research Questions

To address the problem outlined above, this dissertation is guided by the following four research questions:

RQ1 (Understanding):

Which product management activities are most critical in software startups?

This question identifies the activities that have the highest impact on startup survival and growth, distinguishing essential practices from secondary ones.

RQ2 (Compare):

How do product management activities in software startups differ from those in mature organizations?

By comparing contexts, this question clarifies whether existing frameworks can be adapted to startups or whether unique approaches are required.

RQ3 (Prioritization criteria):

Which criteria are most relevant for prioritizing product decisions in software startups?

This explores decision-making under uncertainty, focusing on criteria such as time to value, confidence, and return on investment, and how these shape prioritization in survival-oriented environments.

RQ4 (From insights to tools):

How can evidence-based insights into activities and criteria be translated into actionable frameworks and tools for startups?

This final question ensures that findings are not only theoretical but also practically applicable, offering startups guidance to reduce risk and increase their probability of success.

1.6. Research Methodology

Overarching Research Paradigm and Design

This dissertation adopts an emergent, multi-method research design in which later phases were deliberately shaped by the findings of earlier ones rather than pre-specified at the outset [36-38]. This reflects an honest methodological choice: in a domain as poorly understood as product management in software startups, pre-determining the full research programme would have risked imposing a direction on findings that had not yet been generated. Instead, the research began by exploring what product management looks like and how it is structured in practice — without assuming which aspects would prove most consequential. The finding that activity profiles are similar across startup and mature contexts, but that the consequences of wrong decisions differ fundamentally, was not anticipated at the outset. It was this finding that directed attention toward prioritization logic as the critical differentiator — and toward the need for empirical evidence on which criteria matter and why. Once that evidence had been generated across multiple complementary studies, sufficient grounding existed to synthesize it into a practical tool. The progression from exploration to empirical investigation to prescriptive method development therefore reflects the natural logic of a cumulative research programme, not a pre-planned sequence.

This emergent logic sits within a broader pragmatic methodological stance [36]: rather than adhering exclusively to a positivist or interpretivist paradigm, the dissertation selects methods based on the nature of each research question. Where questions require measurement and

comparison across populations, the design adopts a post-positivist, quantitative orientation. Where questions require understanding of contextual mechanisms and practice-emergent phenomena, the design adopts an interpretivist, qualitative orientation [39, 40]. The result is a sequential exploratory mixed-methods design in which descriptive and quantitative studies establish the empirical foundations, qualitative and mixed-method studies deepen the understanding, and design science research translates the accumulated evidence into a prescriptive tool.

Phase 1, Exploration:

What does product management look like, and does context shape the role?

The first phase addresses RQ1 and RQ2 through two complementary studies. The unit of analysis in both studies is the individual product manager — their activities, attention distribution, and role concentration — rather than the company or startup as a whole. This individual-level focus is appropriate because the research questions concern what product managers do and how they allocate attention, not what outcomes their companies achieve.

Research study 1: Defining the Product Manager Role: A Systematic Literature Review and Organizing Framework of Product Manager Activities.

This study reports a systematic literature review of the product manager role, synthesizing 26 peer-reviewed studies published since 1983. The review consolidates 456 activity mentions into 107 normalized activity concepts and highlights substantial terminological inconsistency across the literature. Building on these consolidated activities, the study introduces an evidence-grounded organizing framework that structures product manager responsibilities into six domains and thirty categories. However, the analysis reveals an uneven distribution of scholarly attention: research is heavily concentrated on strategy, planning, and requirements-related activities, whereas domains such as sales, marketing, and customer success remain sparsely examined. The findings expose a fragmented and inconsistently defined representation of product management in academic research and demonstrate the need for a coherent, activity-based framework to support cumulative theory development.

Research study 2: Measuring Role Concentration in Product Management: An Attention-Based Analysis Using the Herfindahl-Hirschman Index.

This study introduces a quantitative methodology for measuring how software professionals allocate attention across multidimensional role demands, using a dataset of 126 product managers from 37 countries. Grounded in the attention-based view of the firm, the study operationalizes attention as a forced-distribution of 100 time-allocation points across six product management domains derived from the Pragmatic Framework for Product Management. The Herfindahl–Hirschman Index (HHI), traditionally used in economics to assess market concentration, is adapted as a ratio-scale metric that captures the structure of individual role concentration in a single interpretable value. The study details the full methodological workflow—including construct operationalization, survey instrument design, domain calibration, threshold specification, and equivalence-testing procedures—establishing HHI as a rigorously defined, reusable, and empirically robust metric for quantifying attention distribution in software engineering research.

Phase 2, Empirical Deep Dive:

What drives the difference that matters?

The second phase addresses RQ2 and RQ3 through three complementary studies. The unit of analysis shifts at this phase: Studies 3 and 5 examine software startups as cases — examining how each startup's validation behaviour and prioritization criteria relate to performance outcomes. Study 4 returns to the individual practitioner as the unit of analysis, enabling large-scale comparison of prioritization criterion weighting across organizational contexts. This deliberate shift is justified by the research questions: startup-level patterns require company-level analysis, while criterion importance requires individual-level data.

Research study 3: Driving Startup Success Through Hierarchy of Validation.

This study proposes the Hierarchy of Validation framework, designed to guide early-stage software startups in selecting the most appropriate validation activity based on their maturity and risk profile. Through 17 semi-structured interviews with founders and product managers, the study confirms the framework's relevance and usability, particularly for first-time founders facing high uncertainty. The findings show that structured validation choices can reduce waste and improve confidence in product direction.

Research study 4: What Really Matters in Requirements Prioritization: Evidence from 171 Practitioners in Software Startup and Mature Company Contexts.

This study presents a large-scale empirical investigation of requirements prioritization, gathering data from 171 practitioners across 37 countries and 51 industries. The survey instrument was developed through iterative refinement, including AI-assisted pre-testing, expert review, and a practitioner pilot, before respondents were randomly assigned to evaluate prioritization criteria within either a startup or mature company context. The study employs a mixed-methods design combining structured categorical responses, open-ended qualitative inputs, and a hybrid inductive–deductive coding process supported by multi-coder validation. Logistic regression, proportional context scoring, and cross-category consistency analysis are used to quantify contextual differences in prioritization categories and criteria. Through this methodological approach, the study provides one of the most comprehensive and systematically designed datasets to date for examining how requirements prioritization decisions differ across organizational contexts.

Research study 5: Financial and Temporal Decision Criteria in Software Startup Prioritization: Introducing Time to Value (TtV) and Reassessing ROI through Founders' Insights vs. Systematic Literature Review

This study integrates a structured literature review of 40 peer-reviewed studies with a multiple-case study of 34 founders across 19 software startups to examine how requirements-prioritization criteria are conceptualized and operationalized in early-stage environments. The literature review identifies 82 distinct criteria, revealing substantial conceptual fragmentation and limited coverage of financial and temporal decision factors. To address this gap, the study employs a multi-method research design consisting of protocol-driven evidence extraction, cross-case coding, and iterative construct alignment between scholarly models and founder-reported practices. Through this triangulated methodology, the study introduces Time to Value (TtV) as a previously unmodeled temporal criterion and outlines a replicable process for synthesizing prioritization criteria across theoretical and empirical sources. The resulting methodology provides a structured foundation for developing financially and temporally informed prioritization frameworks tailored to high-uncertainty software startup contexts.

Phase 3, Synthesis: Translating findings into a practical tool

Research study 6: VECTR: A Lightweight Requirements Prioritization Method for Software Startups Grounded in ROI, Time-to-Value, and Confidence

This study introduces VECTR, a lightweight requirements prioritization method tailored for software startups and grounded in three empirically relevant criteria: Return on Investment (ROI), Time-to-Value (TtV), and confidence. Rather than proposing new constructs, the contribution lies in operationalizing these dimensions specifically for startup decision contexts and integrating them into a single visual framework suitable for high-uncertainty, resource-constrained environments. The method was developed through prior empirical research and evaluated using a design science approach involving 17 semi-structured interviews with founders, product managers, and technical leads. Practitioners assessed the clarity, relevance, and contextual fit of VECTR after reviewing explanatory materials and a proof-of-concept visualization. Findings indicate that while ROI, TtV, and confidence are already considered in practice, VECTR offers a more explicit, structured, and intuitive way to align decisions with financial and temporal constraints. The study positions VECTR as a practical, startup-appropriate framework whose validation demonstrates perceived usefulness rather than measurable performance outcomes.

Table 1 summarises the methodological positioning of each study. The design is not uniform across studies: it shifts deliberately in response to what each phase requires. Phase 1 (Studies 1–2) is exploratory and confirmatory in nature — using inductive qualitative synthesis and deductive quantitative measurement respectively — because the first task is to establish what product management looks like in practice and whether the role differs structurally across contexts. Phase 2 (Studies 3–5) deepens this by shifting the unit of analysis from the individual product manager to the startup as a whole, combining exploratory case studies with a large-scale quantitative survey and a mixed-method design, in order to identify which validation behaviours and prioritization criteria actually drive different outcomes. Phase 3 (Study 6) is prescriptive: having established the empirical foundations across Phases 1 and 2, the design shifts to abductive reasoning and qualitative evaluation to develop and assess a practical tool. The progression from inductive exploration to deductive confirmation to prescriptive synthesis reflects the cumulative logic of the overall research programme, with each study building on the findings of those preceding it.

Table 1: Methodological positioning of each study

Study	Phase	Orientation	Reasoning	Qual./Quant.	Unit of Analysis
1	Exploration	Exploratory	Inductive	Qualitative	Activities in literature
2	Exploration	Confirmatory	Deductive	Quantitative	Individual PM (n=126)
3	Empirical Deep Dive	Exploratory	Inductive	Qualitative	Startup (n=16)
4	Empirical Deep Dive	Confirmatory	Deductive	Quantitative	Individual PM (n=171)
5	Empirical Deep Dive	Exploratory	Inductive + deductive	Mixed	Startup (n=19) + practitioner
6	Synthesis	Prescriptive	Abductive	Qualitative evaluation	Individual practitioner (n=17)

1.7. Threats to Validity

No research design is without limitations, and it is important to acknowledge the potential threats to validity that accompany this dissertation. These threats do not undermine the value of the findings but highlight the boundaries within which conclusions should be interpreted.

Construct validity

Product management is a broad and multifaceted discipline, encompassing more than one hundred distinct activities across domains such as strategy, development, marketing, and customer success. Capturing this complexity in surveys and interviews inevitably involves simplification. While care was taken to base constructs on prior literature and iterative validation with experts, there remains a risk that some nuances were overlooked or that respondents interpreted terms differently.

Internal validity

Survey and interview participants were asked to reflect on their own practices and perceptions, which may not always align with actual behavior. For instance, respondents might overemphasize activities they perceive as important in theory, while underreporting routine but impactful tasks. Triangulation across multiple methods was employed to mitigate this risk, but it cannot be fully eliminated.

External validity

While the studies involved a substantial number of respondents and cases, the findings are still shaped by the specific contexts in which the participants operate—predominantly software startups, often within European and North American ecosystems. As such, caution is warranted when extrapolating the conclusions to other industries, geographies, or types of startups.

Reliability

Interviews capture rich, contextual insights, but interpretation inevitably involves researcher judgment. To strengthen reliability, interviews were transcribed, coded systematically, and cross-validated with a second reviewer where possible.

1.8. Dissertation Structure

This dissertation follows a paper-based structure comprising eight chapters. This introductory chapter (Chapter 1) has outlined the research context, the knowledge gap and problem statement, the four research questions, and the overarching research methodology and design. Chapters 2 through 7 consist of individual research papers, the majority of which have already

been published in international peer-reviewed venues. Each chapter explores a specific step of the overarching research agenda, building progressively toward the empirically grounded tools (PFPM and VECTR) developed for startup product managers and founders.

Chapter 2 introduces the PFPM through a systematic literature review synthesizing 26 studies and 107 normalized PM activities across six domains and thirty categories [41].

Chapter 3 introduces HHI as a ratio-scale concentration metric applied to 126 product managers across 37 countries, comparing startup versus mature company role concentration.

Chapter 4 presents the Hierarchy of Validation: an inductive, case-based study of 29 semi-structured interviews with 16 startup founders revealing that pre-MVP evidence level, not validation quantity, is the key differentiating factor between higher- and lower-performing startups.

Chapter 5 reports a large-scale survey of 171 practitioners across 37 countries demonstrating statistically significant differences in requirements prioritization criterion weighting between startup and mature company contexts.

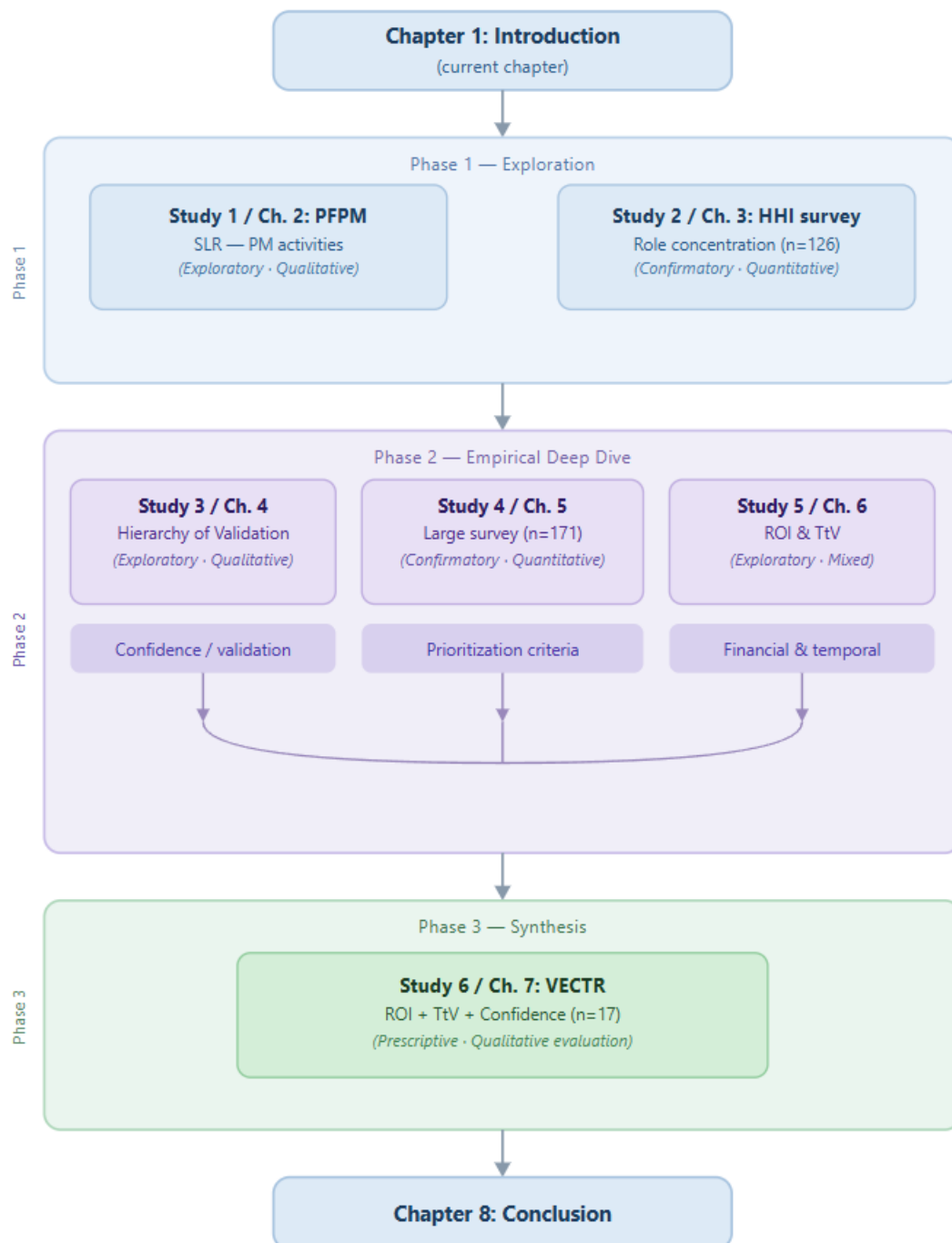
Chapter 6 integrates a structured literature review of 82 prioritization criteria with longitudinal founder interviews across 19 startups, introducing Time to Value (TtV) as a novel, practice-emergent criterion absent from the academic literature.

Chapter 7 introduces VECTR — a lightweight startup-native requirements prioritization method integrating ROI, TtV, and Confidence — developed and evaluated through early-phase artifact evaluation with 17 practitioners.

Chapter 8 concludes the dissertation by synthesizing the key findings contributions, practical implications, limitations, and directions for future research.

The three research phases and the progression from description to prescription are visualised in Figure 1. Studies 1 and 2 map and measure the product management role empirically, establishing what product management looks like in practice and how startup and mature contexts compare. The shift to Studies 3–6 reflects a deliberate widening of scope: once it became clear that activity profiles are similar across contexts but that the consequences of wrong decisions differ fundamentally, the research focus shifted toward the decision logic that drives those decisions — which validation evidence justifies building, which prioritization criteria matter most, and how these insights can be synthesized into a startup-appropriate practical tool. This progression from description to prescription is visualised in Figure 1.

Figure 1: Storyline of the studies used in the dissertation



2. Defining the Product Manager Role: A Systematic Literature Review and Organizing Framework of Product Manager Activities

Abstract¹

Product managers play a central role in software-intensive organizations, yet research provides limited and fragmented insight into what product managers actually do in practice. Although product management is frequently referenced across a wide range of studies, the activities that define the product manager role are often treated implicitly, inconsistently, or in isolation. To address this fragmentation, we conduct a systematic review of 169 empirical and conceptual studies that examine product manager activities and ultimately include 26 studies for analysis. Using an inductive synthesis at the activity and category level, we identify 107 normalized activity concepts. These activities are subsequently organized into an evidence-grounded framework through a higher-order structuring of categories (30) into six empirically grounded domains, namely Executive Leadership, Product Strategy & Planning, Engineering & Development, Sales, Marketing, and Customer Support & Success. The domain-level structure is informed by an established product management reference framework used as an a-priori organizing scaffold and adapted to align with the unit of analysis, rather than imposed deductively. Our synthesis reveals a highly uneven distribution of scholarly attention, with research concentrating on a narrow subset of strategic and planning-oriented activities, while many activities related to coordination, execution, and post-release decision-making receive limited systematic consideration. By organizing extant research around the activities that constitute the product manager role, this study brings coherence to a fragmented literature, clarifies the scope of product management work as represented in academic research, and identifies directions for future research.

Keywords

Product manager role, Product manager activities, Product management, Systematic literature review, Organizing framework

2.1. Introduction

Software-intensive organizations operate in environments characterized by rapid change, technological uncertainty, and strong competitive pressure [24]. In such contexts, product-related decisions have a substantial influence on organizational performance, market success, and long-term viability [24, 47]. Prior research shows that insufficient strategic planning [48], inefficient requirements selection [49], inadequate resource allocation [48], and limited market research and business case analysis [50] can significantly undermine product outcomes and increase the risk of failure. These challenges have been widely documented across studies examining product development, management, and decision-making in software contexts [24, 47, 50].

Within this setting, the Product Manager (PM) is commonly described as the role responsible for guiding product-related decisions across the product lifecycle, including product strategy, planning, coordination, and monitoring of results [23, 48]. Empirical studies associate effective product management practices with improvements in resource efficiency [51], business growth [52], budget control and predictability [50], user satisfaction [3], and release stability [3, 51]. Despite this acknowledged importance, the PM role remains weakly theorized and inconsistently defined in the academic literature [53]. Rather than offering a coherent view of the role, prior research tends to emphasize selected responsibilities—most notably requirements-related activities—while leaving the broader scope of product manager activities fragmented and underexplored [53, 54].

¹ Authors: Frédéric Pattyn and Petra Andries

Recent systematic reviews have begun to synthesize evidence on product manager responsibilities and activities, revealing a large and heterogeneous set of activities discussed across the literature [53, 54]. However, these reviews also indicate that scholarly attention is unevenly distributed: a small number of activities—particularly those related to product planning and requirements—are frequently addressed, while a substantial proportion of identified activities appear only sporadically [53]. This imbalance suggests significant gaps in cumulative knowledge development. However, while these reviews catalogue activities and highlight imbalance in research attention, they do not provide a consolidated organizing structure that systematically integrates activities across domains or resolves terminological fragmentation at the activity level.

As a result, existing SLRs primarily expose *that* fragmentation exists, but leave open the question of *how* the fragmented activity landscape can be coherently structured to support cumulative theory building and comparative research. In particular, prior reviews tend to report activity lists or frequency distributions, without introducing a reusable, evidence-grounded framework that normalizes overlapping activities, clarifies relationships between strategic, operational, and market-facing work, and enables consistent positioning of future studies.

To address this fragmentation, the present study synthesizes findings from a systematic review of 169 empirical and conceptual studies published since 1983 and introduces an evidence-grounded organizing framework for product manager activities: the Pragmatic Framework for Product Managers (PFPM). Building on inductively identified and normalized activities, PFPM consolidates these activities into intermediate categories and subsequently structures them into higher-order domains, providing an explicit organizing lens rather than an additional activity inventory. The PFPM is not intended as a prescriptive process model, but as an integrative lens that clarifies the activities associated with the product manager role and highlights areas of imbalance and underrepresentation in existing research.

By combining a comprehensive synthesis of product manager activities with an organizing framework grounded in prior literature, this study contributes to product management research in three ways. First, it consolidates fragmented insights into a coherent overview of activities examined across the literature. Second, it exposes systematic imbalances in scholarly attention, pointing to underexplored areas of product management work. Third, it provides a structured framework that supports cumulative theory development and serves as a foundation for future empirical research on the product manager role.

2.2. Product Management as a Fragmented Activity Domain

Product management has attracted increasing scholarly attention over the past decades, reflecting its growing importance in software-intensive organizations [24, 47]. This because the role of the Product Manager (PM) has emerged as a driving crucial factor of company success [4, 6, 55]. Despite this interest, research on product management remains conceptually fragmented and lacks a coherent understanding of the activities that define the product manager role [23, 48]. Rather than being examined as a unified domain, product management is frequently addressed indirectly through related research streams such as requirements engineering, new product development, innovation management, and agile software development [49-51]. As a result, product manager activities are often discussed implicitly, inconsistently labeled, or treated as secondary to other focal constructs [52].

A substantial portion of the literature focuses narrowly on requirements-related responsibilities, including requirements elicitation, prioritization, and validation [29]. These activities are commonly examined within the requirements engineering domain, where the product manager role is either subsumed under broader stakeholder roles or reduced to a limited set of decision-making tasks [56, 57]. While this stream has generated valuable insights into requirements-oriented practices, its dominance has contributed to a skewed representation of product

management, in which other activities—such as strategic alignment, market interaction, and cross-functional coordination—receive comparatively limited scholarly attention [49, 51].

Beyond requirements engineering, product management activities appear intermittently across research on innovation management, portfolio decision-making, and organizational coordination [50, 58]. In these contexts, product managers are often portrayed as boundary spanners, integrators, or coordinators between organizational units, yet the specific activities associated with these roles are rarely articulated in a systematic or comparable manner [1, 26]. Consequently, similar activities are described using different terminology across studies, while distinct activities are occasionally grouped under broad or ambiguous labels, further exacerbating conceptual fragmentation [52, 59].

This fragmentation is reinforced by the absence of a commonly accepted structure for organizing product manager activities. Existing frameworks and role descriptions tend to emphasize normative responsibilities or best-practice recommendations, often derived from practitioner-oriented sources, without being explicitly grounded in a systematic synthesis of academic evidence [60]. While such frameworks provide useful guidance for practice, their limited empirical grounding and inconsistent scope make it difficult to align them with the diverse ways in which product management is examined across academic domains [60].

Recent systematic reviews have begun to address this limitation by cataloging product manager activities discussed in the literature [53, 54]. These reviews reveal that product management research exhibits a pronounced long-tail distribution: a small number of activities—primarily related to planning and requirements—appear repeatedly across studies, while a large proportion of activities are mentioned only once or twice [53]. This pattern suggests that the field lacks cumulative focus and that many aspects of the product manager role remain underexplored or weakly theorized [54].

Taken together, these observations indicate that product management constitutes a fragmented activity domain characterized by uneven scholarly attention, inconsistent terminology, and limited conceptual integration [52]. Addressing this fragmentation requires an organizing lens that consolidates dispersed insights, clarifies the scope of product manager activities, and provides a shared foundation for cumulative research. The following sections use a systematic synthesis of the literature to organize these activities into a coherent framework grounded in existing academic evidence.

2.3. Methodology

To consolidate and organize academic knowledge on product manager activities, we conducted a systematic literature review (SLR). A systematic review is appropriate when prior research is dispersed across multiple domains and lacks a cumulative structure, and when synthesizing existing evidence is the primary objective rather than generating new empirical data.

Data Source and Search Strategy

The literature search was carried out using Web of Science (WoS), a multidisciplinary citation index frequently employed in systematic reviews due to its comprehensive coverage of peer-reviewed journals in management, engineering, and information systems. The search was designed to retrieve studies explicitly addressing the activities or responsibilities of product managers. The following query was used:

ALL=((“product manager” OR “productmanager” OR “product management”) AND (activities OR tasks OR responsibilities))

The query was applied to titles, abstracts, and keywords to maximize recall while ensuring conceptual relevance to the product manager role. The search covered publications from 1983 onward, reflecting the emergence of software product management as a distinct research and practice domain. No restrictions were placed on research domain, methodological approach, or

publication outlet at the search stage, reflecting the interdisciplinary nature of product management research. Only peer-reviewed, published academic studies indexed in Web of Science were considered; practitioner literature, books, dissertations, preprints, and grey literature were excluded to ensure consistency and quality of the evidence base.

Screening Procedure

The initial search returned a broad set of publications. To refine the dataset and ensure relevance to product manager activities, we applied the inclusion criteria outlined in Table 1 and the exclusion criteria in Table 2. Following the approach of Hujainah, Bakar [53], we adopted a minimum quality threshold of 50% for study inclusion, corresponding to a score of at least 1.5 out of 3 on our assessment scale for the inclusion criteria. Conversely, studies not scoring 3 out of 3 on the exclusion criteria were removed from the dataset.

Table 2: Inclusion criteria

	Inclusion criteria	1	0,5	0
1	Reporting on expected value.	Explicitly	Superficially	Not mentioned
2	Reporting on the activities (processes).	In-depth	Superficially	Not mentioned
3	Addressing involved roles and titles within the domain.	Product manager explicitly mentioned	Involved roles and titles are mentioned.	Not mentioned

Table 3: Exclusion criteria

	Exclusion criteria	1	0,5	0
1	Year of publication before 1983	≥ 1983	N/A	Before 1983
2	There is no PDF file available	PDF available	N/A	No PDF found
3	Publications are in English	Yes	N/A	No

Acceptance and Quality Criteria

After the studies had been screened for relevance, they were also screened based on quality. In software engineering research, all levels of evidence are typically accepted in systematic reviews. The only commonly suggested threshold is the exclusion of level-5 evidence when sufficient primary studies of higher quality are available. Level-5 evidence refers to sources such as expert opinion, editorials, or anecdotal reports that lack empirical validation². Because

² In software engineering research, evidence levels are commonly adapted from evidence hierarchies proposed by Kitchenham et al. Level-5 evidence typically includes expert opinion, narrative discussion, editorials, or experience reports without systematic data collection or analysis. When sufficient empirical studies (e.g., case studies, surveys, experiments) are available, such sources are often excluded from systematic reviews to strengthen the reliability of synthesized findings.

the primary sources in this review are academic, peer-reviewed publications rather than practitioner books or magazines, this requirement is met by default and is therefore not considered as a separate criterion [61].

Table 3 summarizes the final Quality Assessment Criteria (QAC) applied in this review. The selected criteria are grounded in prior systematic literature reviews in software engineering and requirements engineering, and reflect commonly used dimensions for assessing study rigor and relevance. Specifically, the criteria were chosen to capture (1) scholarly impact, (2) reporting depth, (3) methodological transparency, (4) adequacy of empirical grounding, and (5) clarity of reported results. Furthermore we applied an arithmetic scoring to evaluate study quality as been applied in prior systematic reviews in software engineering, such as those by Hujainah, Bakar [53] and Achimugu, Selamat [62]. In the current review, each criterion is scored on a scale of 0–2, resulting in a maximum possible score of 10. Studies scoring below 5—representing less than 50% of the total possible score—were excluded from the dataset.

Table 4: Quality assessment criteria

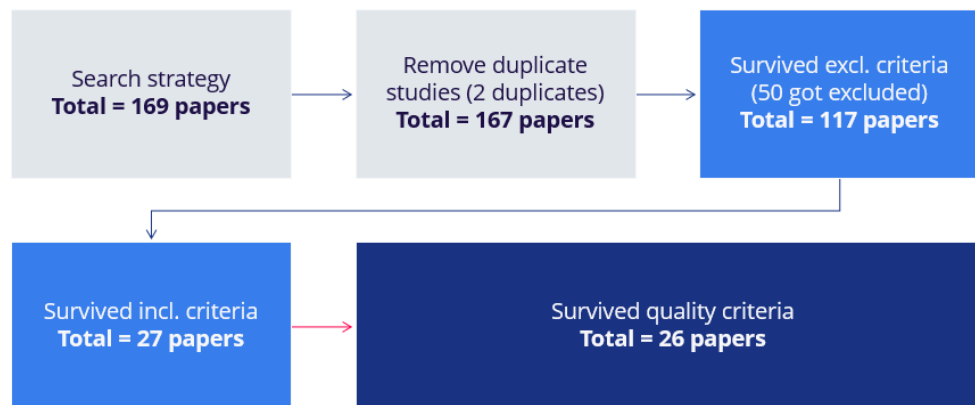
	Exclusion criteria	2	1	0
QAC1	Citations	Citations ≥ 100 (top 2%)	Citations between 100 AND 5	Citations < 5 (lower 50%)
QAC2	Number of references	#ref ≥ 25	#ref between 25 AND 10	#ref < 10
QAC3	Is the proposed methodology (technique/solution) clearly explained?	Yes	Moderate	No
QAC4	Is the evaluation of proposed technique performed on adequate case studies, subjects or project data sets?	Subjects > 10	Subjects between 10 and 4	Subjects < 4
QAC5	Is the result of the study clearly stated?	Yes	Moderate	No

Each criterion operationalizes a distinct quality dimension commonly discussed in the software engineering literature and was scored independently to reduce the risk of excluding methodologically diverse but relevant studies.

Search Results

The initial search retrieved 169 publications from World of Science (WoS). After removing two duplicate records, 167 papers remained for screening. Applying the exclusion criteria (Table 2) resulted in 117 eligible papers. Of these, 27 met the inclusion criteria (Table 1). Finally, quality assessment based on the criteria in Table 3 yielded a total of 26 studies for analysis. Figure 3 illustrates the filtering process and resulting dataset.

Figure 2: Search, screening and acceptance process



Data Extraction and Coding

In a first step, explicit mentions of product manager activities were extracted for each included study. Activities were captured verbatim when possible. To address terminological variation, activities were normalized and consolidated through an iterative coding process. This process focused exclusively on identifying and consolidating activity-level concepts, without introducing higher-order abstraction. The coding proceeded in two analytically distinct stages:

1. Open coding to extract and record all distinct textual expressions referring to product manager activities, tasks, or responsibilities. At this stage, *distinct* refers to unique expressions as reported in the source studies, not yet to conceptually unified activities.
2. Axial coding to normalize these extracted expressions by grouping conceptually equivalent or overlapping items and reconciling synonymous terminology. This step reduced redundancy (e.g., “requirements elicitation” vs. “requirements gathering”) and resulted in a consolidated set of normalized activity concepts, but did not yet involve grouping activities into categories or domains.

Accordingly, this yields a normalized set of activity-level concepts. This hybrid inductive–deductive approach allowed recurring activity patterns to emerge from the literature while maintaining close grounding in the original sources.

Framework Synthesis

In a second step, the coded activities were organized into higher-order categories and domains to create an initial organizing framework. This step represents the first level of higher-order abstraction in the analysis. The purpose of synthesis was descriptive rather than prescriptive: the framework reflects how the product manager role is represented across existing research rather than defining optimal practices.

The synthesis followed a structured progression from activities to categories and from categories to domains. While the identification and normalization of activities were conducted inductively, the structuring at the domain level was informed by an established product management reference framework used as an a-priori organizing scaffold and adapted to fit the unit of analysis (as explained in detail below). Detailed multi-step validation procedures from prior work were intentionally not reproduced; instead, consolidation focused on conceptual coherence and terminology alignment.

Expert Validation

Finally, to assess the plausibility and clarity of the resulting structure, the initial domain and category groupings were reviewed by two independent experts with extensive experience in software product management. Their role was confirmatory: they evaluated whether activities were grouped coherently and whether domain labels corresponded with practitioner

understanding. Minor refinements to terminology and category boundaries were made based on this feedback. This step increased confidence in the interpretive validity of the framework without overextending beyond the scope of evidence synthesis.

2.4. Findings: The Landscape of Product Management Activities

The systematic review identifies a broad and heterogeneous set of activities associated with the product manager role across the analyzed literature. Despite variation in terminology and scope, the 26 studies that met all screening and quality criteria consistently attribute responsibility for a wide range of strategic, operational, and market-oriented tasks to product managers.

Distribution of Activities Across Studies

Across the 26 included studies, we identified 456 mentions of product manager activities. Substantial terminological variation was observed across the reviewed studies. Conceptually similar product manager activities were frequently described using different labels, such as “requirements elicitation” versus “requirements gathering,” or “roadmap creation” versus “roadmap update.” In addition, several activities were reported using broad or ambiguous constructs, including terms such as “stakeholder management” or “strategic alignment,” which encompassed multiple underlying forms of work.

This variation required systematic normalization to consolidate overlapping activity expressions into coherent activity concepts and ensure comparability across studies, as described in Section 3.5 (“Data Extraction and Coding”). This resulted in 107 normalized activity concepts, which form the empirical basis for the subsequent analysis. The distribution of activity mentions is highly skewed: a small set of responsibilities related to monitoring product outcomes, managing requirements, coordinating stakeholders and release cycles, driving innovation, and supporting go-to-market efforts appear frequently, while many other activities occur only once or twice. A complete mapping of the 107 activities’ mentions and the studies in which they appear is provided in Table 4.

Table 5: Distribution of identified activities across publications

Activities	Mentions	Studies
Monitor & control results	17	[12, 63-67]
Requirements analysis	14	[2, 4, 12, 63, 66, 68-72]
Stakeholder management	14	[4, 12, 63, 65, 68, 69, 73-75]
Research and development	14	[2, 4, 63, 66]
Go to market (GtM)	14	[2, 4, 12, 66, 73, 76]

Product lifecycle management	14	[63, 64, 66, 67, 77, 78]
Product strategy & vision	14	[4, 12, 63, 64, 66-68, 79, 80]
Product ideation	13	[4, 63, 66]
Requirements management	12	[2, 4, 12, 16, 68, 69, 72]
Create roadmap	12	[2, 4, 12, 16, 79, 81]
Product design (incl. UI/UX)	11	[4, 63, 64, 66, 74, 75, 80]
Strategic planning	11	[4, 12, 63, 66, 69]
Release management	11	[2, 4, 12, 67, 68, 76, 79]
Product validation	10	[2, 63, 66, 69, 75]
Market research	9	[2, 4, 66, 69]
Project management	9	[2, 4, 12, 63, 74]
Requirements prioritization	9	[2, 12, 63, 79]
Process management (incl. automations)	9	[2, 4, 12, 63, 66]
Quality assurance	8	[63, 66, 75]
Business case analysis	8	[63, 66, 69, 79]
Risk management	7	[4, 12, 14, 69, 73]

Manage software development	7	[4, 12, 14, 69, 73]
Product research	7	[2, 4, 12, 66, 67, 80]
Requirements elicitation	6	[12, 69, 73, 77, 81]
Prototyping	6	[12, 63, 66, 75]
Lead	6	[4, 12, 63, 73, 74]
Positioning	6	[2, 4, 69, 75]
Product planning (incl. releases)	6	[2, 4, 16, 68, 69]
Requirements gathering	5	[4, 12, 68, 71, 79]
Stakeholder communication	5	[4, 12, 69, 74, 75]
Portfolio management	5	[4, 66, 77]
Backlog grooming	5	[12, 72, 81]
Budget management	5	[4, 63]
Sales strategy	4	[2, 75, 77, 80]
Ethical strategic direction	4	[63]
Inspire	4	[4, 63, 65]
Marketing strategy	4	[2, 4, 66]

Marketing planning	4	[65, 75]
Value chain management	4	[64, 80]
Environmental scanning	3	[4, 63, 65]
Update roadmap	3	[64, 65, 69]
Pricing	3	[2, 4, 65]
Marketing research	3	[65, 66]
Write user stories	3	[63, 69, 74]
Collect customer feedback	3	[12, 63]
Resource management	3	[4, 69]
Approve development	3	[12, 66, 77]
Strategic communication	3	[12, 63]
Data analysis	3	[63, 64, 69]
Financial management (incl. funding)	3	[2, 4, 65]
Communication	3	[2, 4, 65]
Evaluate business case	3	[12, 66]
Competitive analysis	3	[4, 12, 82]

GDPR and data privacy	3	[63]
Competitive research	3	[2, 69, 82]
Product value proposition	2	[4, 12]
Marketing copy	2	[65, 79]
Innovation management	2	[69, 80]
Distribution management	2	[2, 4]
Requirements selection	2	[2, 79]
Write product initiation document	2	[4, 66]
Requirements validation	2	[63, 74]
User research	2	[2, 12]
Advertising planning	2	[2, 65]
Forecasting	2	[12, 69]
Create business case	2	[12, 66]
Strategic management	2	[12, 63]
Manage data quality	2	[63]
Tactical planning	2	[4, 69]

Sales execution	2	[2, 4]
Update strategic goals	2	[12]
Product marketing	2	[4, 74]
Define control criteria	2	[63, 65]
Service management	2	[4, 69]
Recruitment	2	[4, 69]
Advertising execution	2	[2, 65]
Release validation	1	[70]
Supplier management	1	[69]
Defect management	1	[4]
Brand management	1	[4]
Use scenarios	1	[4]
Evaluate new requirements	1	[68]
Legal and IP rights management	1	[63]
Create product backlog	1	[12]
Development budgets	1	[2]

Packaging	1	[65]
Marketing communication	1	[4]
Partnership management	1	[69]
Continuous learning	1	[63]
Requirements re-prioritization	1	[12]
Data training (cfr. AI)	1	[63]
Performance management	1	[12]
Manage data acquisition	1	[63]
Brand planning	1	[2]
Approve roadmap	1	[69]
Market research communication	1	[4]
Branding execution	1	[2]
Marketing budget	1	[65]
Resource allocation	1	[12]
Build debt (technical) awareness	1	[12]
Data bias prevention (cfr. AI)	1	[63]

Marketing execution	1	[4]
Data model fairness (cfr. AI)	1	[63]
Cost estimation	1	[83]
Data model transparency (cfr. AI)	1	[63]
Sales planning	1	[75]
Negotiate requirements	1	[71]

Building on the extracted activities summarized in Table 4, the following section introduces an evidence-grounded organizing framework that consolidates these activities into coherent domains.

2.5. An Organizing Framework for Product Manager Activities

As explained above, the activity concepts obtained were then structured into higher-order categories and domains.

From Normalized Activities to Categories

The 107 normalized activity concepts were first grouped into 30 categories based on shared objectives or closely related processes. This categorization step represents the first level of abstraction beyond individual activities and was performed inductively, grounded in recurring patterns observed across the reviewed studies.

Categories were introduced to consolidate activity concepts that referred to similar forms of work while preserving distinctions where activities addressed substantively different concerns. For example, activity concepts related to requirements elicitation, gathering, validation, and prioritization were grouped into a common category, whereas activities related to roadmapping, release planning, and lifecycle coordination were assigned to distinct categories reflecting their differing scopes.

This categorization step provided a necessary intermediate structure between individual activities and higher-order domains, enabling systematic consolidation without prematurely imposing a domain-level framework.

From Categories to Domains using an Organizing Scaffold

The 30 categories resulting from the inductive grouping process were subsequently assigned to higher-order domains to form the final organizing framework. At this level, an established product management reference framework—the ISPMA® Software Product Management Framework (V2.0) [84]—was used as an a-priori organizing scaffold to structure the domains. The ISPMA framework was selected because it provides a comprehensive and widely recognized domain structure for software product management. However, because the ISPMA framework is defined at the level of product management as a discipline rather than at the level

of individual product manager activities, adaptations were required to align it with the unit of analysis used in this study.

Categories were therefore mapped onto ISPMA domains where conceptual alignment was observed. In cases where the ISPMA domain structure did not adequately reflect the empirical patterns identified in the activity data, domain labels were adapted to better fit the observed activity composition. These adaptations did not alter the underlying category or activity groupings, but clarified domain boundaries and terminology.

Resulting Domain Structure

The resulting framework comprises six domains: Executive Leadership [85], Product Strategy & Planning, Engineering & Development [86], Sales, Marketing, and Customer Support & Success [87]. Together, these domains provide a structured representation of how product manager activities are distributed across organizational contexts in the reviewed literature.

Table 5 presents an overview of the relationship between the original ISPMA domain structure and the adapted domains used in this study. Tables 6 to 11 illustrate the internal composition of each domain in terms of categories and associated activities concepts.

The resulting framework thus reflects the empirical activity patterns identified in the literature, structured through a transparent progression from activities to categories and from categories to domains.

Table 6: ISPMA versus the Pragmatic Framework for Product Managers (PFPM) domains

ISPMA	PFPM domains
Strategic Management	Executive leadership
Product Strategy	Product Strategy & Planning
Product Planning	Product Strategy & Planning
Development	Engineering & Development
Marketing	Marketing
Sales & Fulfilment	Sales
Delivery Services & Support	Customer Support & Success

Table 7:Executive leadership domain: categories and activities

Category	Activity
Strategic management & communication	Strategic management
	Strategic planning
	Strategic communication
	Stakeholder management
	Partnership management
	Stakeholder communication
Supplier, legal & IP management	GDPR and data privacy
	Legal and IP rights management
	Supplier management
Human resource management	Continuous learning
	Performance management
	Recruitment

Corporate strategy & vision	Ethical strategic direction
	Lead
Financial management (incl. funding)	Budget management
	Financial management (incl. funding)
Monitor & control results	Monitor & control results
Portfolio management	Portfolio management
Approve roadmap	Approve roadmap

Table 8:Product strategy & planning domain: categories and activities

Category	Activity
Product strategy & vision	Inspire
	Product strategy & vision
	Update strategic goals
Product lifecycle management	Product lifecycle management
Business cases	Build debt (technical) awareness
	Business case analysis
	Cost estimation
	Create business case
	Development budgets
	Evaluate business case
	Forecasting
	Risk management
	Write product initiation document
Roadmapping	Create roadmap
	Update roadmap
Product value proposition	Product value proposition
Product backlog management	Backlog grooming
	Create product backlog
Product planning (incl. releases)	Go to market (GtM)
	Product planning (incl. releases)
	Release validation
Requirements management (incl. prioritization, selection...)	Data analysis
	Environmental scanning
	Evaluate new requirements
	Innovation management
	Negotiate requirements
	Process management (incl. automations)
	Product design (incl. UI/UX)
	Product ideation
	Product research
	Project management
	Requirements elicitation
	Requirements gathering
	Requirements management
	Requirements prioritization

	Requirements re-prioritization
	Requirements selection
	Use scenarios
	Value chain management
	Write user stories
Resource management	Resource allocation
	Resource management

Table 9: Engineering & development domain: categories and activities

Category	Activity
Research & development	Prototyping
	Research and Development
	Data bias prevention (cfr. AI)
	Data model fairness (cfr. AI)
	Data model transparency (cfr. AI)
	Data training (cfr. AI)
	Manage data acquisition
	Manage data quality
Manage software development	Manage software development
	Tactical planning
	Define control criteria
	Requirements analysis
Quality assurance	Product validation
	Quality assurance
	Approve development
	Defect management
	Requirements Validation
Release management	Release management

Table 10: Sales domain: categories and activities

Category	Activity
Pricing	Pricing
Sales execution	Sales execution
	Sales planning
	Sales strategy
Distribution management	Distribution management

Table 11: Marketing domain: categories and activities

Category	Activity
Market research & communication	Competitive analysis
	Competitive research
	Market research
	Market research communication
	Marketing research
	User research
Positioning	Positioning
Branding	Brand management
	Brand planning
	Branding execution
	Packaging
Marketing execution	Marketing budget
	Marketing communication
	Marketing copy
	Marketing execution
	Marketing planning
	Marketing strategy
	Product marketing
Advertising	Advertising execution
	Advertising planning

Table 12: Customer support & success domain: categories and activities

Category	Activity
Customer support	Service management
	Collect customer feedback
	Communication

Each of the six framework domains encompasses multiple activity categories, grouping conceptually related responsibilities identified in the systematic review. The domains differ substantially in size, reflecting variation in how frequently different areas of product management are addressed in the reviewed literature.

As shown in Tables 6 to 11, activities are most densely concentrated in the Product Strategy & Planning domain (n = 42). Executive Leadership and Engineering & Development each comprise 19 activities, while the Sales and Customer Support & Success domains contain comparatively fewer activities. Marketing occupies an intermediate position in terms of activity count.

This distribution highlights how product manager activities identified in the literature are unevenly spread across organizational domains, with a strong concentration around strategy, planning, and development-related work.

2.6. Discussion

Positioning and Scope of the Framework

The organizing framework presented in this study is grounded explicitly in the evidence synthesized from the systematic literature review and reflects how product manager activities are represented across existing academic research. It is therefore descriptive rather than prescriptive in nature. The framework does not define best practices, prescribe organizational ownership, or imply that all product managers perform all identified activities. Instead, it captures the breadth of responsibilities attributed to the product manager role in the literature, acknowledging that role enactment varies across organizational contexts, including differences in organizational size, maturity, industry, and product domain.

By structuring normalized activity concepts into six domains, the framework provides a coherent representation of the fragmented activity landscape reported across studies. Importantly, this structure should not be interpreted as a normative role model. Rather, it reflects how research has collectively described product manager work, making visible both areas of concentration and areas that remain underrepresented. This positioning aligns with the study's objective to organize and clarify existing knowledge rather than to evaluate effectiveness or prescribe organizational arrangements.

The deliberate descriptive scope of the framework allows it to function as a stable reference point for future research. Because it does not assume uniform role definitions or organizational structures, it can support comparative analysis across contexts—such as startups versus mature organizations—without imposing assumptions about how product management should be practiced.

Uneven Scholarly Attention Across Product Manager Activities

The Results demonstrate a pronounced imbalance in how product manager activities are represented in the academic literature. Activities related to product strategy, planning, and development coordination dominate existing research, while activities associated with sales enablement, pricing, customer success, and cross-functional commercialization receive comparatively limited attention. This uneven distribution reflects patterns in scholarly focus rather than an assessment of the practical importance of these activities.

The framework makes these imbalances explicit by organizing activities across domains and reporting their relative representation. As such, it highlights where cumulative knowledge has emerged and where empirical investigation remains sparse. This observation is particularly relevant for understanding the boundaries of current product management research, as it suggests that substantial portions of the product manager role are under-theorized despite their relevance in practice.

By surfacing these gaps systematically, the framework provides a structured basis for prioritizing future research efforts. Rather than expanding the activity inventory further, subsequent studies can use the framework to direct attention toward underexplored domains and to examine how neglected activities contribute to product and organizational outcomes.

Fragmentation and Terminological Ambiguity in Product Management Research

A further contribution of this study lies in making explicit the extent of terminological fragmentation across product management research. As reported in the Results, conceptually similar activities are frequently labeled using different terms, while other responsibilities are described using broad or ambiguous constructs that obscure underlying distinctions. This fragmentation complicates comparison across studies and limits the accumulation of shared constructs.

The need for extensive normalization during the coding process illustrates how definitional ambiguity persists even for core product management activities. This observation helps explain why the product manager role has resisted consolidation into a unified theoretical foundation

and why prior research often treats product management as a loosely defined set of responsibilities rather than as a coherent activity system.

By normalizing activity concepts and organizing them within a consistent structure, the framework addresses this fragmentation at an analytical level. While it does not resolve all definitional disputes, it provides a common reference that can support clearer positioning of future studies and reduce ambiguity in how product manager activities are described and compared.

Implications for Research and Practice

Taken together, the findings suggest that progress in product management research is constrained less by a lack of empirical studies and more by the absence of integrative structures that support cumulative knowledge development. The framework introduced in this study responds to this limitation by offering an evidence-grounded organizing lens that connects fragmented study streams without imposing a prescriptive model.

The framework can serve as a foundation for future research in several ways. First, it enables systematic comparison of research emphasis across domains and activity categories. Second, it provides a basis for examining context-dependent role configurations, such as differences between startups and large enterprises or between consumer-facing and enterprise software contexts. Third, it offers a starting point for empirical validation efforts aimed at examining how different clusters of product manager activities relate to performance, coordination, or organizational outcomes.

Importantly, these avenues for future work build on the framework's descriptive scope rather than extending it into normative territory. This restraint preserves analytical flexibility and ensures that the framework remains applicable across diverse research contexts.

Limitations and Directions for Future Research

This review synthesizes published academic literature and therefore reflects the biases inherent in that corpus. Activities that are widely practiced in industry but underrepresented in peer-reviewed research may be absent or appear marginal. The reliance on terminology-based search criteria also risks excluding studies in which product manager responsibilities are embedded implicitly under broader roles, such as project managers, founders, or product owners [SLR]. Although expert validation mitigated interpretive bias in category assignment, further empirical work is needed to assess whether the domain and category structure aligns with practitioner reality across diverse organizational settings.

Future research can expand on this work along several dimensions. First, comparative studies are needed to investigate how product manager activities differ across organizational contexts—particularly between startups and mature companies. Given differences in resource constraints, decision-making structures, and market uncertainty, the breadth and prioritization of activities may diverge significantly across these environments. Understanding how contextual factors shape role enactment would contribute to more nuanced theory and support tailoring frameworks such as PFPM to specific settings.

Second, empirical validation through surveys, interviews, case studies, or ethnographic research can test whether the activities and domains identified in this synthesis align with actual practice. Such validation would help determine whether the PFPM reflects enacted responsibilities or primarily captures academic representations.

Finally, the uneven scholarly attention observed in the review highlights clear gaps for future inquiry. Commercialization, customer success, pricing, value proposition development, and cross-functional sales integration are central to product success in practice yet remain weakly theorized. Deeper exploration of these areas would strengthen the conceptual grounding of

product management and expand the field beyond its historical focus on planning and requirements work.

2.7. Conclusion

This study consolidates and structures academic knowledge on product manager activities through a systematic review of 26 peer-reviewed studies and the identification of 107 normalized activity concepts. The Results show that scholarly attention is unevenly distributed across the activity landscape, with research concentrating primarily on activities related to product strategy & planning, and research & development, while activities associated with executive leadership, sales, marketing, and customer support receive comparatively limited attention. These patterns reflect not only substantive gaps in empirical coverage but also broader conceptual fragmentation in how the product manager role is represented across the literature.

To address this fragmentation, the Pragmatic Framework for Product Managers (PFPM) organizes the identified activities into six empirically grounded domains and associated categories. Together, these domains provide a coherent structure that captures the full range of product manager activities reported in prior research, without privileging a particular process model or organizational arrangement. By normalizing divergent terminology and making patterns of scholarly attention explicit, the framework supports clearer conceptual boundaries and enables systematic comparison across previously disconnected study streams.

The study makes three contributions to product management research. First, it offers an evidence-based consolidation of product manager activities into categories and domains, clarifying the scope and diversity of responsibilities attributed to the role in existing scholarship. Second, it exposes systematic imbalances in research attention across domains, identifying underexplored areas that warrant further empirical investigation. Third, it introduces a descriptive organizing lens that facilitates cumulative theory building by providing a shared structure for positioning and integrating future studies.

Future research is needed to empirically validate the framework across organizational contexts, examine how role enactment varies with factors such as organizational size and maturity, and deepen understanding of underrepresented domains including commercialization, pricing, and customer success. By establishing a structured evidence base and revealing persistent patterns of fragmentation, this study lays the groundwork for more coherent, cumulative, and context-sensitive research on the product manager role.

3. Measuring Role Concentration in Product Manager Role: Using the Herfindahl-Hirschman Index: Comparison of Startups and Mature Companies.

Abstract³

The Product Manager role increasingly spans multiple functional domains, yet research lacks validated metrics to quantify how they distribute attention across competing responsibilities. We introduce the Herfindahl-Hirschman Index (HHI)—traditionally used to measure market concentration in economics—as a metric for individual-level role concentration. HHI captures whether attention is concentrated in few domains (specialist) or distributed across many (generalist), providing a single interpretable score. We demonstrate HHI's application through an empirical study of 125 product managers across 37 countries, testing whether organizational context (startup vs. mature company) shapes concentration patterns across six functional domains. We find no significant difference in role concentration between contexts (Startup $M=0.247$ vs. Mature $M=0.255$, $p=.662$, $d=0.08$), with 94% of respondents having generalist profiles ($HHI < 0.40$) and spending 30% of time on Product Strategy & Planning. These findings demonstrate HHI's utility for studying role concentration in work patterns.

Keywords: Product Management, Startups, HHI, Role Concentration, Attention Allocation

3.1. Introduction

Product management in software engineering is increasingly characterized by multidimensional responsibility structures that span strategy, development, coordination, and market-facing activities [88]. Rather than occupying a narrowly defined technical or managerial role, product managers are expected to engage concurrently across a broad range of functional domains — from executive leadership and product strategy to engineering coordination, marketing, sales, and customer success [4, 42, 84]. Understanding how product managers distribute their limited time and attention across such competing demands has significant implications for role design, team composition, hiring practices, and organizational effectiveness.

Despite this complexity, software engineering research lacks quantitative metrics to capture how product managers should structure their attention across multidimensional role demands. Existing studies typically examine activity domains [13] or contexts [7] independently—through importance ratings, frequency rankings, Likert scales, or regression-based domain comparisons [89]—without measuring overall concentration or diversification patterns. This analytical approach reveals which activities practitioners emphasize but misses a crucial structural dimension: whether attention is concentrated in a narrow subset of domains or distributed broadly across many.

This measurement limitation is theoretically consequential. The attention-based view [90] (ABV) of organizations posits that decision makers cannot and will not pay equal attention to all the tasks they are expected to perform [91]. Because attention is structurally channeled through roles, routines, and communication processes, variation in performance and decision outcomes may depend not only on which activities are emphasized, but also on how attention is distributed across them. However, empirical studies in software engineering that analyze activity domains independently cannot capture this structural dimension. Two individuals may allocate identical time to a dominant activity while differing substantially in how remaining time is distributed across other domains—yet domain-level comparisons would not detect this fundamental difference in role configuration.

Consider a concrete example. Two product managers each allocate 40% of their time to product strategy. One distributes the remaining 60% relatively equally across five other domains (approximately 12% each); the other concentrates remaining time heavily in engineering

³ Authors: Frédéric Pattyn, Petra Andries and Peter Goetz

coordination (35%) with minimal attention to marketing, sales, and customer success (5-10% combined). Domain-level analysis focused on strategy emphasis would treat these individuals as equivalent. Yet their role configurations differ fundamentally: one maintains a diversified generalist profile while the other combines strategy focus with engineering specialization. Without a metric that captures the overall distribution of attention, such structural differences remain invisible.

To address this gap, this study introduces the Herfindahl-Hirschman Index [92] (HHI) as a measure of individual-level role concentration. Originally developed in industrial organization economics to assess market concentration [93, 94], HHI captures the extent to which a distribution is concentrated by summing squared proportional shares. When applied to product managers' role analysis, HHI provides a single interpretable metric that reflects the structural concentration of product managers' attention across multiple domains. In contrast to domain-level means, which reveal first-order allocation patterns, HHI captures second-order structural patterns—whether attention is evenly distributed over a broad range of domains or concentrated within a subset of them.

Interestingly, the ABV proposes that decision-makers' attention to selected issues and activities depends on structural characteristics of the organization, including the firm's rules, resources and social relationships [90]. The distinction between a start-up context of mature organization [27] can be expected to be of particular relevance in this respect [45, 51]. It is well-known that established firms and start-ups differ widely with respect to size, resource availability, and formalization of routines and relationships [24]. According to the ABV [95], it is not unlikely that these characteristics lead to a differential distribution of product managers' attention.

Practitioner discourse often claims that product managers must “wear many hats” and act as broad generalists across business functions [96]. Yet, in both contexts, there are strong arguments for more focused roles. In mature organizations, product managers can afford to specialize deeply [20, 97], whereas in startups their limited resources make prioritization essential—often requiring them to specialize out of necessity [51, 98].

Existing research has documented variation in perceived importance across domains [89], yet (a) whether such differences translate into structural concentration patterns, and (b) whether these concentration patterns differ between start-up and mature contexts remains untested. By applying HHI to proportional time allocations across six domains, this study enables systematic comparison of role concentration between organizational contexts.

This paper therefore addresses two research questions:

RQ1: Can the Herfindahl-Hirschman Index provide a valid and interpretable measure of individual-level role concentration in software engineering roles?

RQ2: Does organizational context (startup versus mature firm) influence structural role concentration among product managers?

The contribution of this paper is primarily methodological. By introducing a concentration metric grounded in attention theory and demonstrating its empirical applicability, we provide software engineering researchers with a validated tool for examining attention allocation across multiple roles. The empirical findings regarding product management serve as a demonstration of this tool's utility while also yielding substantive insights that challenge common assumptions about context-dependent role specialization.

The remainder of this paper is structured as follows. Section 2 elaborates the theoretical background, including the attention-based view and the measurement challenge in studying multidimensional roles. Section 3 details the research methodology, including the operationalization of HHI for role concentration. Section 4 presents empirical results. Section 5 discusses theoretical and methodological implications, including guidance for applying HHI to other software engineering roles. Section 6 concludes.

3.2. Background

Multidimensional Activity Domains in Product Management

Product management in software engineering requires individuals to allocate limited cognitive and temporal resources across a broad range of competing functional domains. Prior research, including a systematic review of the product management literature [96], identifies six core domains that together define the scope of the role: Executive Leadership, Product Strategy & Planning, Engineering & Development, Marketing, Sales, and Customer Support & Success [42].

Product managers occupy a particularly boundary-spanning position, engaging across all six of these domains simultaneously — formulating strategic vision and managing roadmaps, coordinating with development teams on requirements and delivery, supporting commercial activities such as pricing and sales enablement, and managing customer-facing responsibilities including onboarding and feedback collection. Their responsibilities thus extend well beyond the technical domain to encompass market analysis, competitive positioning, and broader organizational communication. Whether product managers should concentrate their attention in a narrow subset of domains or distribute it broadly across many domains, remains unclear.

This question may be even more pressing in startup contexts, where these responsibilities unfold under high environmental uncertainty [50] and resource constraints [99], increasing prioritization pressure [45, 46] and sharpening attention trade-offs [100].

The Attention-Based View

The attention-based view [93] (ABV) provides a theoretical lens for understanding role configuration as attention structure. ABV conceptualizes organizations as systems in which decision outcomes depend on how attention is distributed across issues and alternatives. Three core principles underpin this perspective.

First, decision-makers possess bounded cognitive capacity [101]. They cannot attend to all issues simultaneously; instead, attention to one issue necessarily reduces attention available for others. This scarcity principle means that attention allocation involves trade-offs rather than independent decisions about each domain.

Second, organizational structures channel attention [102]. Roles, routines, communication patterns, and formal procedures direct attention toward particular activities and away from others. A product manager's role definition, reporting relationships, and meeting cadence all shape which issues receive attention and which are neglected.

Third, the distribution of attention—not merely its total quantity—shapes outcomes [103]. Two organizations (or individuals) may devote equal aggregate attention to product development, yet differ in how that attention is distributed across different aspects of product development. These distributional differences may produce divergent decisions and outcomes.

For individual role analysis, ABV implies that role configuration should be understood as attention structure. A role is not simply a list of activities but a pattern of attention allocation across activities. This pattern may be concentrated (attention focused on few activities) or diversified (attention spread across many). The structural pattern matters analytically because it reflects prioritization under constraint and may predict outcomes such as coordination effectiveness, decision quality, and individual performance.

Moreover, the ABV explains that decision-makers' attention structure will depend on structural characteristics of the organization, including the firm's rules, resources and social relationships [90]. Organizational rules and procedures shape how attention is distributed by directing decision-makers toward a limited or broad set of issues, thereby concentrating or diffusing their focus. Similarly, the availability of resources and the configuration of social relationships influence whether attention is narrowly concentrated on a few priorities or spread across a broader range of concerns.

Although the relevance of attention structure has been well-established in the management domain [90], empirical research in software engineering rarely operationalizes attention as a structural distribution. Studies of product management frequently rely on activity importance ratings, rankings, or regression analyses at the domain level. Such approaches reveal which domains are prioritized on average but do not capture whether attention is concentrated in a narrow subset of domains or diversified across many.

The Measurement Challenge

The absence of concentration metrics creates several analytical limitations.

First, structural differences between individuals remain undetected. Domain-level analysis treats two individuals with identical mean allocations to a focal domain as equivalent, even if their overall profiles differ substantially. The example in the introduction—two product managers with 40% strategy allocation but different distributions of remaining time—illustrates this limitation.

Second, contextual comparisons become incomplete. If organizational context (e.g., startup versus mature) shapes role specialization, one would expect observable differences in overall concentration patterns rather than only marginal shifts in domain means. Without a concentration metric, such structural variation remains unobservable.

Third, theoretical claims about generalism and specialism lack empirical grounding. Practitioner discourse distinguishes "generalist" from "specialist" role configurations [13], yet these categories remain impressionistic without quantitative operationalization.

To address these limitations, a metric is required that: (1) captures the overall concentration of a multidimensional allocation vector; (2) enables comparison across contexts and individuals; (3) produces interpretable values on a meaningful scale; and (4) remains grounded in theoretical frameworks for understanding attention. The Herfindahl-Hirschman Index satisfies these requirements.

The Herfindahl-Hirschman Index

The Herfindahl-Hirschman Index measures concentration by summing squared proportions across categories [93]. In industrial economics, HHI quantifies market concentration [94, 95]: markets where few firms hold large shares produce high HHI values (concentrated), while markets with many firms holding similar shares produce low values (competitive). The index has been applied extensively in antitrust analysis [104, 105], with established interpretation thresholds developed through decades of regulatory practice.

Formally, for a distribution across n categories with proportional share p_i in category i :

$$HHI = \sum_{i=1}^n p_i^2$$

The squaring transformation has important properties. It gives greater weight to large shares than to small ones—increasing a 30% share by 10 percentage points raises HHI more than

increasing a 5% share by the same amount. This property aligns with intuitions about concentration: a distribution with one dominant category and many small ones is more concentrated than a distribution with several medium-sized categories, even if the largest category is similar in size.

HHI is bounded between $1/n$ (perfect diversification, equal shares across all categories) and 1.0 (complete concentration in a single category). For a six-category distribution, these bounds are 0.167 and 1.0 respectively.

We adapt HHI to measure individual-level role concentration. When applied to time allocation across role domains, HHI captures whether an individual concentrates attention on few domains (high HHI, specialist profile) or distributes attention across many (low HHI, generalist profile). The metric provides a single interpretable score that reflects structural concentration—complementing domain-level analysis by capturing the overall pattern rather than individual components.

Interpretation Thresholds

Meaningful interpretation requires calibrated thresholds [106]. We adapt thresholds from antitrust economics, where HHI values have established interpretive meaning (Table 12):

Table 13: Herfindahl–Hirschman Index (HHI) thresholds

HHI Range	Market Interpretation	Role Interpretation
< 0.15	Highly competitive	Highly diversified (uniform distribution)
0.15–0.25	Moderately competitive	Moderately diversified
0.25–0.40	Moderately concentrated	Moderately concentrated
≥ 0.40	Highly concentrated	Highly concentrated (specialist)

For a six-domain role structure, we operationalize generalist profiles as $HHI < 0.40$ and specialist profiles as $HHI \geq 0.40$. This threshold represents meaningful concentration: achieving $HHI \geq 0.40$ with six domains requires substantial skew toward one or two dominant domains.

To illustrate, consider boundary cases for a six-domain allocation:

- Equal distribution (16.7% each): $HHI = 0.167$
- One domain at 40%, rest equal (12% each): $HHI = 0.232$
- One domain at 50%, one at 30%, rest equal (5% each): $HHI = 0.350$
- One domain at 60%, one at 25%, rest minimal: $HHI = 0.430$
- One domain at 70%, rest distributed: $HHI \geq 0.500$

The 0.40 threshold thus corresponds to configurations where one or two domains clearly dominate the allocation profile.

3.3. Methodology

Research Design Overview

We conducted a cross-sectional survey between November 2023 and January 2024 targeting professionals with active software product management experience. The study employed an experimental vignette design where participants were randomly assigned to evaluate either a startup or mature company scenario and report how they would allocate their time. Only participants whose actual professional experience (as reported in the survey) matched their assigned context were retained in the final dataset, ensuring that all reported time allocations reflect genuine lived experience. A robustness analysis including non-matched participants is reported in Section 4.4 and does not alter the conclusions.

Context Conditions

Participants were presented with one of two context descriptors corresponding to their actual organizational setting:

Startup context: "You are working as a Product Manager in a software startup with the following characteristics: The company is less than 5 years old, has fewer than 50 employees, recently raised Series A or Series B funding (or is bootstrapped), operates with informal processes and flat hierarchy, and faces high uncertainty requiring frequent pivoting."

Mature company context: "You are working as a Product Manager in a mature software company with the following characteristics: The company is more than 10 years old, has more than 100 employees, is either Series C+ funded or profitable, has established processes and defined hierarchies, and operates in relatively stable market conditions."

These context descriptors operationalize the startup/mature distinction along multiple dimensions — age, size, funding stage, process formalization, and environmental stability — rather than relying on a single indicator. This multidimensional operationalization better captures the holistic contextual difference while avoiding debates about which single indicator best defines "startup" status.

Sample and Data Collection

Sampling strategy. Respondents were recruited through multiple channels to ensure diversity: LinkedIn professional networks (62% of final sample), Product Day Conference (20%), International Software Product Management Association (ISPMA) (15%), and Start it @ KBC accelerator program (3%). This multi-channel approach produced a geographically diverse sample spanning 37 countries across North America, Europe, Asia, and other regions.

Data quality and context matching. From 182 initial responses, 57 were excluded based on four criteria: (1) incomplete responses to core survey tasks or failed attention checks; (2) time allocations that could not be rescaled to sum to 100%; (3) context non-matching, where a participant did not have professional experience in the context they were present (startup or mature); and (4) one response excluded due to a data recording error. This yielded a final analytical sample of 125 participants (87 startup, 38 mature) with complete, verified time allocation data suitable for HHI analysis.

Sample characteristics. Table 13 presents demographic characteristics of the final analytical sample.

Table 14: Sample Characteristics

Characteristic	Full Sample (n=125)	Startup (n=87)	Mature (n=38)
Demographics			
Mean age (years)	38.2 (SD=8.1)	37.1 (SD=7.8)	40.6 (SD=8.4)
Gender: Female / Male	29.8% / 70.2%	31.0% / 69.0%	26.9% / 73.1%
Education: $\geq$ Master's degree	63.5%	60.9%	69.2%
Professional Background			
Mean PM experience (years)	6.1 (SD=3.2)	5.8 (SD=3.0)	6.7 (SD=3.6)
Founder experience (≥ 1 startup)	36.8%	42.5%	23.1%
Geographic Distribution			
Belgium	40%	37.9%	43.6%
United States	9.6%	10.3%	7.7%
India	5.6%	6.9%	2.6%
Other (34 countries)	45.2%	44.9%	46.1%
Industry Sector			
Software/SaaS	45.2%	48.3%	38.5%
Technology (non-software)	23.8%	21.8%	28.2%
Non-technology	31.0%	29.9%	33.3%

No significant demographic differences emerged between randomly assigned groups (all χ^2 tests $p > .10$).

Measurement: Time Allocation Across Domains

Domain structure. We operationalized product management activities using a six-domain structure derived from the Pragmatic Framework for Product Management (PFPM) [3], a widely-used industry framework that provides comprehensive coverage of product management responsibilities:

1. Executive Leadership: Strategic vision development, stakeholder management, business case development, executive communication
2. Product Strategy & Planning: Roadmap development, feature prioritization, release planning, product lifecycle management
3. Engineering & Development: Technical coordination, sprint planning, requirements specification, development team collaboration

4. Marketing: Product positioning, messaging, go-to-market strategy, competitive analysis, market research
5. Sales: Revenue enablement, sales support, pricing strategy, customer acquisition support, deal involvement
6. Customer Success: Customer onboarding, retention activities, support escalation, customer feedback collection and synthesis

Forced-distribution measurement. To measure behavioral time allocation rather than perceived importance, we employed a forced-distribution task designed to eliminate response inflation common in independent Likert ratings:

"Distribute 100 points across the six domains below to represent the relative time or attention you would allocate in your assigned context. The total must sum to exactly 100. Think about a typical month in this role."

This measurement approach offers several advantages over alternatives. First, forced distribution captures trade-offs under bounded attention constraints—increasing allocation to one domain necessarily decreases allocation to others, mirroring real resource constraints. Second, it produces ratio-scale data with true zero points and equal intervals, enabling parametric analysis unavailable with ordinal importance ratings. Third, it reduces social desirability bias by focusing on behavioral allocation rather than perceived importance.

Construct validity. We employ time allocation as a behavioral proxy for the theoretical construct of "attention" from the attention-based view. While cognitive attention and behavioral time are not perfectly equivalent—individuals might invest equal time across domains while concentrating cognitive resources differently—time allocation provides a practical and validated approximation.

Operationalization: Computing HHI for Role Concentration

For each product manager, we computed the Herfindahl-Hirschman Index from their six-domain time allocation:

$$HHI = \sum_{i=1}^6 \left(\frac{\%_i}{100} \right)^2$$

where %_i represents the percentage allocation to domain i.

Worked example. Consider a product manager with the following allocation:

- Executive Leadership: 20%
- Product Strategy & Planning: 30%
- Engineering & Development: 40%
- Marketing: 5%
- Sales: 3%
- Customer Success: 2%

$$HHI = (0.20)^2 + (0.30)^2 + (0.40)^2 + (0.05)^2 + (0.03)^2 + (0.02)^2$$

$$HHI = 0.0400 + 0.0900 + 0.1600 + 0.0025 + 0.0009 + 0.0004 = 0.294$$

This product manager shows moderate concentration ($\text{HHI}=0.294$): while 40% of time focuses on engineering coordination, they still meaningfully engage across multiple domains, qualifying as a generalist under our threshold ($\text{HHI} < 0.40$).

Classification. We classified respondents as generalists when $\text{HHI} < 0.40$ and specialists when $\text{HHI} \geq 0.40$. This threshold reflects moderate concentration while preserving sensitivity to meaningful skew, as discussed in Section 2.5.

Scale Type and Validity of Parametric Analysis

Before proceeding to the analytical strategy, we address the measurement-theoretic properties of The Herfindahl-Hirschman Index (HHI) [92] score to justify the use of parametric statistical techniques. Following Stevens [107] taxonomy of measurement scales, a variable qualifies as ratio scale if it satisfies four criteria: (1) a meaningful ordering of values, (2) equal intervals between values, (3) a non-arbitrary (true) zero point, and (4) the permissibility of ratio comparisons. We demonstrate that the HHI as computed in this study satisfies all four.

Table 15: Ratio-scale properties of the HHI score as applied in this study

Property	How HHI satisfies it
(1) Meaningful order	HHI values are strictly ordered: a higher HHI score unambiguously indicates greater concentration than a lower one. A product manager with $\text{HHI} = 0.60$ is more concentrated than one with $\text{HHI} = 0.25$, regardless of which domains dominate.
(2) Equal intervals	HHI is computed as $H = \sum(p_i^2)$, a continuous mathematical function of proportional shares. A difference of 0.10 anywhere on the scale represents the same absolute increase in the sum of squared shares — the intervals are equal by construction, not by assumption. This contrasts with ordinal scales (e.g., Likert items) where intervals between response categories are not guaranteed to be equal.
(3) True zero point	When all n domains receive equal shares ($p_i = 1/n$ for all i), HHI reaches its theoretical minimum of $1/n$ — not zero. However, the scale has a theoretically grounded anchor at both ends: $H = 1/n$ represents the maximum possible equality given n categories, and $H = 1$ represents complete concentration in a single category. Crucially, the zero point in this study is non-arbitrary: it corresponds to a perfectly uniform distribution, which is a meaningful and interpretable state of role allocation. This satisfies the ratio-scale requirement for a true (non-arbitrary) origin.
(4) Ratio comparisons	Because the zero point is meaningful, ratio comparisons are permissible. A product manager with $\text{HHI} = 0.60$ is twice as concentrated as one with $\text{HHI} = 0.30$. This is a substantive claim about role structure, not merely a numerical artefact. Such multiplicative interpretations are possible only on ratio scales.

These properties follow directly from the mathematical construction of HHI and do not depend on any particular application domain. They hold regardless of whether the proportions (p_i) represent market shares, time allocations, or any other exhaustive and

mutually exclusive set of proportions that sum to 1. In the original economics literature, the HHI was explicitly designed as a measure of concentration [92] suitable for arithmetic operations, comparisons of magnitude, and parametric statistical analysis including t-tests, ANOVA, and regression — and this usage is consistent across its applications in industrial organization [108], banking regulation value [109], and behavioural science [110].

Summary of the measurement-theoretic argument: The input to HHI — time allocation proportions from a forced-distribution task — is ratio scale by design (each $p_i \in [0, 1]$, with a true zero at 0% and a meaningful unit at 1%). The HHI transformation $H = \sum(p_i^2)$ is a smooth, monotonic, continuous function applied to these ratio-scale inputs, preserving the ordering and interval properties. The result is a derived ratio-scale score with a mathematically defined range $[1/n, 1]$, equal intervals by construction, and a non-arbitrary minimum. Parametric statistical techniques — specifically the independent samples t-test and Cohen's d effect size — are therefore appropriate and their assumptions (continuous outcome, meaningful mean differences, equal intervals) are satisfied.

As an additional empirical check, we verified that the distribution of HHI scores in the sample was approximately normal (Shapiro-Wilk $W = 0.97$, $p = .18$; skewness = 0.42, kurtosis = -0.31), that variance was homogeneous between the startup and mature company groups (Levene's test $F = 1.14$, $p = .29$), and that no floor or ceiling effects were present (minimum observed HHI = 0.174, maximum = 0.821, with 95% of observations between 0.19 and 0.71). These empirical properties confirm that the assumptions underlying the t-test are met in practice, not just in theory.

Analytical Strategy

Given the ratio-scale properties of HHI established above, we used two complementary approaches to address our research questions.

Descriptive analysis. We first examined how product managers allocated time across domains, looking at averages, variation, and ranges. We then examined the distribution of concentration scores, including how they clustered, how much they varied, and how many participants fell into generalist versus specialist categories. This descriptive foundation provides context for the subsequent analyses.

Traditional hypothesis testing. To test whether startup and mature contexts differ, we compared average concentration scores between groups using t-tests. We calculated effect sizes using Cohen's d to assess whether any differences were practically meaningful, not just statistically detectable. We verified that the data met standard assumptions for these tests.

3.4. Results

Domain-Level Allocation Patterns

Table 16 presents descriptive statistics for time allocation across the six product management domains, separately for startup and mature company contexts.

Table 16: Time Allocation by Domain and Context

Domain	Full Sample M (SD)	Startup M (SD)	Mature M (SD)	Diff	t-test p
Executive Leadership	15.9% (10.2)	15.8% (10.4)	16.1% (9.8)	-0.3%	.87

Product Strategy & Planning	29.3% (12.1)	29.1% (12.3)	29.7% (11.7)	-0.6%	.80
Engineering & Development	20.5% (11.8)	20.7% (12.1)	20.0% (11.2)	+0.7%	.76
Marketing	11.2% (8.4)	11.5% (8.7)	10.5% (7.8)	+1.0%	.54
Sales	11.4% (9.2)	11.2% (9.4)	11.8% (8.8)	-0.6%	.74
Customer Success	11.7% (9.8)	11.7% (10.0)	11.8% (9.4)	-0.1%	.96

Several patterns emerge from the domain-level analysis.

First, Product Strategy & Planning dominates across both contexts, receiving approximately 30% of product manager time on average. This substantially exceeds all other domains and confirms the centrality of roadmap and prioritization activities to the product management role.

Second, Engineering & Development receives substantial attention at approximately 20% allocation. This reflects the coordination-intensive nature of product management work, where ongoing collaboration with development teams is essential.

Third, customer-facing domains cluster together at approximately 11-12% each. Marketing, Sales, and Customer Success receive similar average allocations with minimal context differences.

Fourth, context differences are minimal. The largest mean difference across any domain is 1 percentage point (Marketing), far below practically meaningful thresholds. No domain-level differences reach statistical significance.

Fifth, substantial individual variation exists. Standard deviations of 8-12 percentage points indicate considerable heterogeneity in individual allocation patterns within each context. This variation underscores the importance of examining structural patterns beyond group means.

These domain-level findings replicate prior observations that product management roles remain broadly cross-functional across organizational stages. However, as emphasized in Section 2, domain-level means cannot reveal whether overall attention is concentrated or diversified. The following analysis addresses this structural question using HHI.

Distribution of Role Concentration (HHI)

HHI values in our sample ranged from 0.170 (near-perfect diversification) to 0.664 (substantial concentration), with mean 0.249 (SD=0.100) and median 0.227 (Table 17). The distribution was approximately symmetric with slight positive skew, showing no floor or ceiling effects.

Table 17: Distribution of HHI Scores

Statistic	Full Sample	Startup (n=87)	Mature (n=38)
Mean	0.249	0.247	0.255
Standard Deviation	0.100	0.100	0.102
Median	0.227	0.225	0.235

Minimum	0.170	0.170	0.170
Maximum	0.664	0.664	0.460
Interquartile Range	0.190–0.290	0.188–0.287	0.195–0.298

Classification by concentration thresholds. Table 16 presents the distribution of product managers across HHI categories.

Table 18: Classification by Concentration Category

HHI Range	Category	Count	Percentage
< 0.20	Highly diversified	29	23.2%
0.20–0.30	Moderately diversified	72	57.6%
0.30–0.40	Moderately concentrated	19	15.2%
≥ 0.40	Highly concentrated (Specialist)	5	4.0%
< 0.40	Generalist (total)	118	94.0%

Several key findings emerge:

Generalist profiles dominate.

Fully 94% of product managers maintain generalist profiles ($\text{HHI} < 0.40$), with only 5 individuals (4.0%) qualifying as specialists.

Modal pattern is moderate diversification.

The largest segment (57.6%) clusters in the “moderately diversified” range ($\text{HHI} 0.20\text{--}0.30$). This suggests that most product managers distribute their time unevenly—typically with Product Strategy absorbing ~30%—while still engaging meaningfully across multiple domains.

Extreme specialization is rare.

Only 5 product managers show $\text{HHI} \geq 0.40$, and just one exceeds 0.50. The maximum observed HHI (0.664) is a substantial outlier; the 95th percentile lies near 0.45. These cases likely represent role configurations shaped by acute constraints (e.g., technical bottlenecks, funding cycles, or GTM pressures), particularly in startups.

Distributions are similar across contexts. Visual inspection of density plots reveals nearly identical distributions for startup and mature conditions, with similar central tendencies, spreads, and shapes. Although overall distributions are similar across contexts, it needs to be noted that four of the five identified specialists appear in the startup condition, suggesting that extreme specialization—while uncommon—may be more likely to occur in a startup context. We look at this in more detail in the following section.

Role concentration in different contexts

We compared role concentration between startup and mature organizational contexts using multiple statistical approaches.

The average concentration scores were nearly identical across contexts. Startup product managers averaged 0.247, while those in mature companies averaged 0.255—a difference of

less than one percent of the possible scale. This trivially small difference carries no practical significance.

Traditional hypothesis testing confirmed no statistical difference between groups. T-tests yielded a p-value of .662, well above conventional significance thresholds, and the confidence interval for the difference included zero within a narrow range. The effect size was negligible by conventional standards, smaller than what would typically be considered even a "small" effect.

The categorical distribution of role profiles was also equivalent across contexts. Comparing the proportion of participants classified as highly diversified, moderately diversified, moderately concentrated, and highly concentrated revealed no meaningful differences between startup and mature companies. Both contexts produced nearly identical distributions across these categories.

All statistical assumptions were verified. Tests for normality and equality of variance indicated the data met requirements for the analyses conducted.

Robustness Analyses

We conducted four robustness checks (Table 19) to assess whether the null finding depends on sample composition or model specification.

Table 19: Summary of Robustness Analysis

Analysis	Comparison	Effect Size (d)	p-value	Interpretation
Full sample (n=125)	Startup vs. Mature	0.08	.662	Negligible difference
Belgian exclusion (n=76)	Startup vs. Mature	0.02	.93	Effect shrinks further
Experience moderation	Context × Experience interaction	$\beta = 0.004$	.78	No moderation
Software/SaaS only (n=57)	Startup vs. Mature	0.08	.77	Identical to full sample
Tech non-software (n=30)	Startup vs. Mature	0.06	.87	Consistent
Non-tech (n=39)	Startup vs. Mature	0.08	.81	Consistent
Full sample incl. non-matched (n=182)	Startup vs. Mature	0.14	0.39	Consistent

Belgian subsample exclusion. Belgian participants represent 40% of the sample, raising potential concerns about geographic concentration. Re-analyzing with Belgian participants excluded (n=76) yields an even smaller effect ($d = 0.02$ vs. $d = 0.08$), suggesting the null finding strengthens when removing geographic concentration.

Experience moderation. Product management experience might moderate context effects—perhaps experienced product managers maintain generalist profiles regardless of context while less experienced ones show context-dependent patterns. Adding an Experience ×

Context interaction term yields a non-significant interaction ($\beta = 0.004$, $p = .78$). The null effect holds across experience levels from early-career through senior practitioners.

Industry subgroups. Role definitions might differ across industries. Separate analyses for Software/SaaS ($n=57$), Technology non-software ($n=30$), and Non-technology ($n=39$) sectors all yield negligible, non-significant context effects. The pattern is consistent across industries.

Full sample including non-matched participants. Re-analyzing with all 182 respondents before applying context-matching criteria yields consistent results (Startup $M=0.240$ vs Mature $M=0.253$, $d = 0.14$, $p = .39$). The null finding is not an artifact of the matching procedure.

Summary of Empirical Findings

The empirical investigation yields four key findings.

First, organizational context does not affect how time is allocated across individual domains. Startup and mature product managers distributed their attention across all six functional areas in virtually the same proportions, with less than one percentage point difference across all domains.

Second, generalist profiles dominate the sample. Ninety-four percent of product managers maintained generalist concentration patterns ($HHI < 0.40$) regardless of whether they worked in startup or mature environments. Still, the extremes are mainly to be found at startups with a focus on Product Strategy & Planning.

Third, this null result is robust. The finding holds across geographic subsamples, experience levels, industry sectors, and whether participants' actual work context matched or differed from their assigned scenario..

According to the findings, the anticipated specialization in either context does not materialize. Although theory suggests that mature organizations should enable deeper specialization and that startups might enforce specialization through sharper prioritization, the evidence shows that product management remains broadly multidisciplinary in both settings. Rather than diverging into specialist and generalist configurations, PMs across organizational maturities continue to span multiple domains, indicating that the role itself resists meaningful specialization.

3.5. Discussion

HHI as a Valid Metric for Role Concentration (RQ1)

The findings demonstrate HHI's utility as a metric for measuring individual-level role concentration in product management role contexts. Several observations support this conclusion.

Interpretable score distribution. HHI values in our sample distributed meaningfully across the theoretical range, with sufficient variance to differentiate individuals and no problematic floor or ceiling effects. The distribution shape was approximately normal, supporting parametric analysis.

Meaningful classification. The predefined thresholds successfully distinguished concentration profiles. The 94% generalist classification aligns with theoretical expectations for a boundary-spanning role like product management, while the small specialist minority (4%) identifies genuine outliers with concentrated profiles.

Captures structural patterns invisible to domain-level analysis. While domain-level analysis revealed that Product Strategy consistently receives ~30% of time—a clear first-order

finding—HHI revealed that this domain emphasis coexists with diversified engagement across remaining domains. The distinction between domain-level skew and structural concentration illustrates HHI's analytical value: a product manager can focus on a dominant domain while remaining structurally generalist.

Grounded in established measurement tradition. HHI's [93] adaptation from economics provides interpretive anchors absent in ad hoc metrics. The extensive application of HHI in antitrust contexts [104] validates its mathematical properties and provides guidance for threshold calibration.

These observations collectively validate HHI as a measurement tool for checking task concentration on software engineering research on multidimensional roles.

Organizational Context and Role Concentration (RQ2)

Although theory suggests that product managers in both startups and mature organizations could be expected to specialize—startups through forced prioritization and mature firms through structural support—we find no meaningful concentration differences between the two settings. This absence of specialization in either context warrants careful interpretation.

Structural similarity despite contextual differences. Startups and mature organizations differ substantially in size, process formalization, resource availability, and environmental uncertainty. Yet these contextual differences do not translate into different role concentration patterns among product managers. This suggests that role structure is shaped more by the inherent demands of product management than by organizational context.

Inherently boundary-spanning role. Product management imposes multidisciplinary demands that persist regardless of context. The role's core function — coordinating across strategy, development, and market functions — requires engagement across domains by definition. The data show that this structural characteristic holds across both startup and mature environments, with domain-level allocations differing by less than one percentage point across all six domains.

Challenging assumptions. This finding challenges expectations derived from both practitioner narratives and theoretical arguments. Even though startups might be expected to drive specialization through forced prioritization, and mature organizations might support specialization through formal structures, our evidence shows no such differentiation. Neither context appears to enable meaningful specialization for product managers. Future research could explore whether this pattern extends to other boundary-spanning roles.

Structural Skew Within Diversification

Although overall concentration is low, domain-level allocations reveal systematic skew. Product Strategy & Planning consistently accounts for approximately one-third of total attention, exceeding other domains in both contexts. This pattern illustrates an important distinction: diversification does not imply uniform distribution.

Roles can be structurally generalist (low HHI, broad engagement across domains) while exhibiting internal hierarchy (some domains systematically receive more attention than others). The product management role exemplifies this pattern: it is generalist in structure (attention distributed across all six domains) but hierarchical in emphasis (strategy dominates, followed by engineering, with customer-facing domains receiving less).

This observation suggests a multi-level analytical opportunity. Future research might apply concentration metrics within dominant domains to examine intra-domain allocation patterns. For example, within the Product Strategy domain, one could examine whether attention is

concentrated on roadmapping versus prioritization versus release planning. Such nested concentration analysis would further operationalize attention-based theory by examining structure within attention channels.

Applying HHI to Other Software Engineering Roles

While demonstrated using product management, the HHI approach is transferable to other multidimensional software engineering roles such as engineering managers, architects, and technical leads. Many such roles are assumed to differ in specialization across organizational contexts, yet empirical measurement of structural concentration remains limited. Applying HHI to proportional time or effort allocations in these roles would enable systematic testing of generalism and specialization patterns. Future studies should carefully justify domain partitioning and calibrate concentration thresholds relative to domain count.

Theoretical Implications

The findings carry implications for attention-based theorizing in software engineering contexts.

Operationalizing attention structure. By translating attention-based concepts into quantifiable metrics, this study provides a template for empirical ABV research in software engineering. The attention-based view has been influential in management theory but underutilized in software engineering research, partly due to operationalization challenges. HHI addresses this gap by providing a measure consistent with ABV's emphasis on structural allocation patterns.

Generalism as structural feature. The near-universal generalist profiles (94%) suggest that generalism may be a structural feature of certain roles rather than a context-dependent outcome. Product management appears to require broad engagement across domains regardless of organizational characteristics. Future research could examine whether this structural generalism predicts outcomes such as coordination effectiveness or career progression.

Practical Implications

The findings also carry implications for practice.

Role design. The evidence suggests that product management roles require generalist configurations regardless of organizational size or maturity. Organizations designing PM roles should anticipate broad engagement across strategy, development, and commercial domains rather than expecting context-appropriate specialization.

Hiring and assessment. Selection criteria emphasizing breadth of experience and cross-functional capability appear relevant across contexts. The assumption that startup contexts require "scrappier" generalists while mature organizations need domain specialists lacks empirical support.

Career development. Product managers should expect multidimensional demands throughout their careers rather than progressive specialization. Career development might focus on deepening capability across domains rather than narrowing scope.

3.6. Limitations

Several limitations bound the inferences from this study.

Self-reported allocation. Time allocation was measured via self-report, which may diverge from actual behavior. Retrospective time estimation is subject to known cognitive biases, and the degree of correspondence with actual behavioral patterns remains to be established for

this domain structure. Future research should triangulate with behavioral measures such as calendar data or experience sampling where feasible.

Domain granularity dependence. HHI values depend on how domains are defined. Our six-domain structure provides theoretical coverage but alternative partitions might yield different absolute values. We expect relative patterns (between-context comparisons, generalist/specialist classifications) to remain stable if domains are conceptually coherent, but this assumption warrants testing.

Cross-sectional design. We captured concentration at a single point. Organizational roles may evolve as companies scale or as products mature. Longitudinal designs could examine whether individual-level concentration patterns shift over time and whether such shifts relate to organizational changes.

Single role demonstration. The empirical validation focuses on product management. Generalizability to other software engineering roles remains to be established. The methodological contribution (HHI as a tool) is role-agnostic, but empirical patterns may differ across roles.

Context matching and sample balance. The requirement that participants' actual professional experience match their assigned context descriptor ensures ecological validity but results in unequal group sizes (87 startup, 38 mature). The smaller mature group limits statistical power for detecting small effects. Future research should aim for more balanced samples across organizational contexts, ideally through targeted recruitment of mature-company product managers.

Sample characteristics. The sample, while geographically diverse, shows Belgian concentration (40%) and may not represent global PM populations. Software/SaaS representation (45%) exceeds other industries. Robustness checks suggest results hold across subgroups, but replication in different populations would strengthen confidence.

3.7. Conclusion

This paper introduced the Herfindahl-Hirschman Index as a quantitative metric for measuring individual-level role concentration of product managers in software engineering. Grounded in attention-based theory, HHI operationalizes structural attention allocation by capturing how effort is distributed across multiple functional domains within a role. By translating attention structure into a single interpretable metric with established interpretation thresholds, the study provides software engineering researchers a validated tool for examining multidimensional roles beyond domain-level comparison.

Using product management as an empirical demonstration, we find that role concentration remains structurally low and consistent across startup and mature organizational contexts. Ninety-four percent of product managers maintain generalist profiles regardless of organizational setting. The combination of traditional hypothesis testing and robustness analyses provides converging evidence that context does not shape structural concentration — challenging common assumptions about context-dependent role specialization and suggesting that product management imposes inherently multidisciplinary demands that persist across organizational maturity stages.

The methodological contribution extends beyond product management. HHI provides a principled approach for examining whether attention is concentrated or diversified in any role characterized by multidimensional responsibilities. As software engineering work becomes increasingly boundary-spanning—with developers, architects, engineering managers, and

other roles navigating complex responsibility structures—tools for measuring structural patterns become increasingly valuable.

Future research might extend this approach in several directions: applying HHI to other software engineering roles to assess generalizability; examining whether concentration predicts outcomes such as performance, satisfaction, or coordination effectiveness; investigating temporal dynamics in concentration patterns; and triangulating self-reported allocations with behavioral measures such as calendar data or experience sampling to establish the correspondence between perceived and actual attention distribution.

By introducing a concentration metric grounded in established measurement traditions and demonstrating its empirical applicability, this study contributes to more structurally precise understanding of role configuration in software engineering research and practice.

4. Validate Before You Build: Exploring Pre-MVP Evidence Levels – Not Quantity – and Startup Performance in Early-Stage Software Ventures

Abstract⁴

Software startups operate in environments characterized by rapid change, high uncertainty, and limited resources, resulting in high failure rates and challenges such as premature scaling and cash flow mismanagement. This study investigates how early-stage software startups validate their initial idea before building their first Minimum Viable Product (MVP). Through 29 semi-structured interviews with founders from 16 software startups, pre-MVP validation activities were extracted and inductively coded into a six-level Validation Canvas spanning three validation stages identified in the literature: Problem Validation, Problem-Solution Fit, and Product-Market Fit. No clear relationship was observed between the number of validation events and startup performance. Instead, stronger-performing startups tended to reach higher levels of evidence — particularly securing contingent investment commitments (Level 5) or paying customers (Level 6) before full MVP development. Level 6 — paying customers before the full product exists — is identified as the strongest form of pre-MVP market evidence, as it directly validates willingness-to-pay without relying on investor confidence. In this study, product-market fit is operationalised as demonstrated commercial viability through external financial commitments rather than interest signals or free sign-ups alone. Based on these exploratory findings, the study proposes the Hierarchy of Validation: a staged, bidirectional process model in which bottom-up traversal from informal interest signals (L1) toward paying customers (L6) emerged as the primary pattern among stronger-performing startups. A top-down direction, in which experienced founders begin at higher evidence levels and work downward, is proposed as a hypothesis for future research. These findings remain exploratory and require validation in larger and more diverse samples.

Keywords:

Startups, MVP, Minimal Viable Product, Requirements Validation, product-market fit, PMF, Validation Canvas, Hierarchy of Validation, Minimal Viable Experiment, MVE

4.1. Introduction

Software startups, notorious for their high pace, uncertainty, and resource limitations [24], face numerous challenges, including premature scaling, cash flow mismanagement affecting 82% of startups, difficulties in securing investment (47%) [50, 98], and simply running out of funds (21 to 44%) [51, 56, 100]. The failure rate stands at 63% [100], with 25% occurring in their first year [99]. The situation is aggravated by an overemphasis on feature-heavy, slow to market [99] Minimum Viable Products (MVP) without clear business strategy or user traction [1]. Operating under significant time and market pressures [25, 111] with limited resources [112, 113], startups must rapidly deliver high-innovation solutions [111], make quick assumptions about the problems they aim to solve, and validate these through MVPs [114]. Many startups fail within the first years due to poor problem-solution fit [115] and neglecting the iterative learning process during MVP development [33, 116], whether it's through Continuous Experimentation (CE) [117] or the Lean Startup methodology [118].

Continuous Experimentation (CE) provides a systematic way [117] to validate an idea by following a rigorous process that eliminates opinion from the discussion [119, 120]. It was first termed by McKinley 2012 [121] at Etsy to assess the Return on Investment of potential features. CE involves the ongoing deployment of new features and enhancements while leveraging customer feedback and usage behavior to guide product development [117]. The key challenges include designing rapid and effective experiments, advanced instrumentation for data collection and analysis, and integrating results into both product roadmaps and

⁴ Authors: Frédéric Pattyn, Usman Rafiq, Frederic Gailly and Peter Goetz

software development processes [117]. Through the years the intensity of the research has been increasing consistently [119]. In the meantime, a family of generic approaches has been proposed [117], for example the Lean Startup method.

The Lean Startup methodology, introduced by Eric Ries, emphasizes the creation of cost-effective MVPs [118]. These failing startups often didn't do sufficient market research or business case analysis [50], leading to inefficient requirements selection process [29], misallocation of critical resources [57], and an over-commitment to solutions that do not meet market needs [122]. Consequently, they face the risk of not achieving product-market fit [57], and struggle with acquiring their first paying customers [24], putting the startup's runway (number of months a company has left until cash runs out) at risk and therefore probability of success. In this challenging financial climate, effective cash flow management, strategic planning, and efficient resource allocation are crucial for startup survival. Increasing the confidence level of early-stage ideas can significantly [3] influence future performance [53, 54], making the validation of ideas through the MVP stage essential to mitigating the risk of entrepreneurial failure [32]. Making crucial financial decisions, from conceptualizing an idea to transforming it into a viable early-stage venture, becomes increasingly important and daunting.

This study involved conducting 29 semi-structured interviews with 16 founders of software startups. These interviews covered their journey from concept to reality, including their first successes and failures. The aim is exploratory: to investigate what types and levels of pre-MVP validation activity are associated with stronger startup performance, and to generate the Hierarchy of Validation as a structured framework for future confirmatory research. The research question explored in this study is:

“How do startups validate their initial idea before building their first MVP?”

4.2. Related work

The concept of the Minimum Viable Product (MVP) was initially introduced by Eric Ries [118], specifically within the context of Lean Startups [123]. The Lean Startup (LS) framework, a highly respected framework within the field of product management, as proposed by Ries [124], emphasizes the importance of creating a cost-effective MVP in a startup context. The goal is to swiftly enter the market to enhance the product's customer value by incorporating user feedback. Continuous Experimentation (CE) could be used to assess which features might be in fact generate user value [117]. This iterative approach makes it possible to progress from problem-solution fit to product-market fit [115]. Designing experiments for the latter stage are more challenging, but preferable. This concise description encompasses many of the challenges and success factors commonly addressed in the startup literature [33].

Startups often strive to achieve a balance between cost efficiency and revenue generation by embarking on the creation a MVP [24, 125]. Addressing this challenge within the realm of early-stage ventures involves defining the MVP's feature set coverage [26, 126, 127], focusing exclusively on the core requirements [24, 125] that are most likely to elicit heightened user satisfaction [30]. The CE experiments should also be focused on these core requirements in order to avoid unnecessary development costs [115]. Given the advantageous synergy associated with these frugal innovations [128], it is reasonable to expect that startups would take this consideration into account when making RPC decisions. Furthermore, the development of an MVP imparts momentum to a team and expedites the time-to-market [129, 130], offering additional benefits. CE therefore can be considered a method for incremental innovation [131]. The MVP process is characterized by rapid experimentation [117] and iterative learning. Initially, startups make assumptions about the problems they aim to solve and quickly develop tangible prototypes to test these assumptions [32, 132, 133]. This experimentation involves a

cycle of building, measuring, and learning, which is central to the Lean Startup methodology [32]. By testing and validating ideas early, startups can avoid investing in products that may prove unsuccessful in the market [116]. This approach mitigates risks and increases the likelihood of finding a viable product-market fit [33].

Developing an MVP comes with several challenges. Startups operate under significant time and market pressures, often with limited resources [112, 113]. The need to deliver high innovation solutions in the shortest possible time while managing small resources is a recurring theme [25, 111]. Other challenges include user interface design, financial constraints, communicating ideas to stakeholders, identifying important features, security concerns, product differentiation strategy, choosing the appropriate market channel, lack of appropriate team members, and selecting the appropriate platform [133]. Additionally, the competence of founders and team members plays a crucial role in prioritizing requirements and validating ideas [116].

Validation is a critical component of the MVP process. Various methods can be employed, including digital prototypes, 3D MVPs, paper prototypes, and landing pages [133]. Validation practices also encompass usability tests, A/B tests [120], usage data analysis, interviews, surveys, trade show testing, and testing with influencers [32, 113, 114, 116, 133, 134]. Challenges in this phase include obtaining enough subjects for testing, fear of contacting customers, technical challenges, and concerns about intellectual property theft. Despite these challenges, early feedback from validation efforts provides a better understanding of customer needs, prioritizes development activities, and avoids unnecessary development efforts [32].

The MVP approach offers substantial value to startups. It allows for early testing of market demands, gathering customer feedback, and testing fundamental business hypotheses [135]. MVPs enable startups to quickly learn from their experiments and refine their products based on real user input. This process not only improves the chances of achieving a successful product-market fit but also accelerates sales to the first customers [133]. Moreover, incubators and accelerators can play a significant role in MVP development by providing the necessary education and support to create the right product-market fit [116]. The overall goal of an MVP is to bring value to customers, gather meaningful feedback, and promote positive word-of-mouth, ultimately leading to a more successful and sustainable startup venture [33, 112].

4.3. Study design

A purposive sampling method [136] was applied to select cases that were theoretically relevant to the research objective, namely understanding validation practices in early-stage software startups operating under high uncertainty. Specifically, startups were selected from the Start it @KBC accelerator program in Belgium, one of Europe's largest startup incubators, which provided access to a comparable cohort of actively developing early-stage ventures. Inclusion criteria required that each case: (i) was a software-intensive startup in an active product development phase; (ii) had at least one founder or co-founder available for interview (see Table 24); and (iii) was still operating in a pre-product-market-fit context, meaning the startup had not yet achieved a stable, repeatable, and self-sustaining commercial model [137]. These criteria were considered important because this study specifically targets early-stage decision-making under uncertainty, where assumptions about customer needs, value propositions, and market demand still require systematic validation (see Table 18). Founders were selected as the primary informants because, in pre-product-market-fit startups, they are typically the direct decision-makers responsible for product prioritization and validation choices [138]. Unlike hired product managers, who may join at a later stage, founders hold first-hand knowledge of the rationale behind early-stage decisions. This makes them the most appropriate and reliable source of data on the phenomena under study.

Restricting the sample to startups within a similar accelerator environment and maturity stage increased contextual comparability across cases while ensuring direct relevance to the research

question. For each of the 16 software startups, a single key informant was interviewed, consisting of either the founder or co-founder (see Table 24). All startups belonged to the same quarterly cohort (Q4/2022). Between Q4 2022 and Q2 2024, a total of 29 semi-structured interviews were conducted, with an average duration of 63 minutes.

Where possible, startups were followed up at approximately six-month intervals over the course of the study. The depth of longitudinal coverage varied across cases. For startups that received three interviews, strong contact and a cooperative relationship made scheduling straightforward. For those with two interviews, follow-up was generally possible but occasionally hampered by time constraints or a reluctance to participate. For startups with only a single interview, follow-up was not possible — either because the company had ceased operations by the time of the next planned contact, or because repeated outreach attempts went unanswered. These follow-up interviews were designed to deepen understanding of decisions already discussed, rather than to track change longitudinally in a strict sense.

We followed the general interview guidelines [139] to ensure consistency and reliability in our data collection. In accordance with our ethics policy, we ensured that all participants and their shared information would remain confidential.

The interview transcripts were analyzed using thematic analysis [140]. First, all interviews were transcribed and read multiple times to increase familiarity with the data and to obtain an overall understanding of recurring patterns related to startup validation practices, uncertainty reduction, and early-stage decision-making. In a second step, initial codes were generated inductively by the primary researcher through systematic line-by-line coding of relevant fragments. These codes were subsequently grouped into broader candidate themes by identifying conceptual similarities and recurring validation behaviors across cases. The coding process was iterative rather than strictly linear, meaning that codes and themes were continuously refined as additional interviews were analyzed. To strengthen reliability and reduce potential single-coder bias, a second coding pass was performed independently using an AI-assisted coding approach grounded in prior research on AI-supported qualitative analysis [141]. The outputs of both coding passes were compared to identify areas of convergence and divergence. Where differences in interpretation occurred, these were reviewed manually and resolved through deliberation until conceptual agreement was reached. Final themes were only retained once stable thematic patterns emerged consistently across the cases. The guiding interview questions used during the semi-structured interviews can be found here: <https://tinyurl.com/34wc6muy>.

In order to study which requirements prioritization criteria or processes could be considered predictive or at least a shared correlative indication among the best performing startups, it's required to rank them. Based on the interviews, grey literature and official records, we've gathered data on (1) number of funding rounds, (2) total funding raised, (3) revenue generated in most recent year, (4) profits generated in most recent year, (5) remaining runway in months and (6) if they are still in business or not.

Table 20. Startups profiles

ID	Description	Interviews	Years active
S1	Football data analytics	1 (1h15)	3
S2	Security platform	2 (2h07)	2
S4	Web3 SocialFi	3 (2h45)	3
S5	PLG platform	2 (1h13)	4
S7	HR tech	1 (0h45)	2
S8	Logistical solution	1 (1h25)	3
S9	Form builder	3 (3h24)	3

S10	Web2 credit market	2 (1h45)	3
S11	HR tech	3 (2h53)	3
S12	Healthcare platform	2 (1h48)	2
S13	HR tech	2 (1h15)	3
S14	HR tech	1 (2h11)	3
S15	Data document management	3 (2h41)	3
S16	Fashion tech	1 (0h58)	2
S17	AI for trends analysis	1 (1h23)	4
S18	HR tech	1 (0h51)	4

Startup performance was assessed (see Table 19) using a composite comparison approach based on multiple indicators collected during the study period, including factors such as funding progression, revenue evolution, organizational growth, operational continuity, and overall business viability at the time of last contact. Startups that had ceased operations (S5, S7, and S14) were treated as a separate lowest-performing category — bankruptcy represents a categorically different outcome from slower growth, and these startups are not ranked against operational ones. The remaining startups were grouped into three tiers: Tier 1 (green: S2, S10, S11); Tier 2 (yellow: S18, S15, S12, S17, S4); Tier 3 (orange: S9, S1, S8, S13, S16). Startups no longer in operation are referred to as 'out of business' (OOB) throughout.

Table 21. Startups and interviews

ID	(1)	(2)	(3)	(4)	(5)	(6)	Final ranking
S2	1	2	3		4	Y	2.5
S10	1	1	1	11	2	Y	3.2
S11	3	5	2	3	5	Y	3.6
S18	3	13	4	1	1	Y	4.4
S15	2	4	5	9	4	Y	4.8
S12	3	9	7		1	Y	5
S17	3	3	9	10	3	Y	5.6
S4	2	6	10		5	Y	5.7
S14	3	7	6	6	11	N	6.6
S5	3	8	10	8	6	N	7
S9	3	11	9	5	7	Y	7
S1	3	10	10	7	8	Y	7.6
S8	4	14	9	2	10	Y	7.8
S13	4	14	8	4	11	Y	8.2
S16	3	12	10		9	Y	8.5
S7	4	14	10		12	N	10

T1=Tier 1 (green), T2=Tier 2 (yellow), T3=Tier 3 (orange), OOB=out of business (red).

4.4. Results

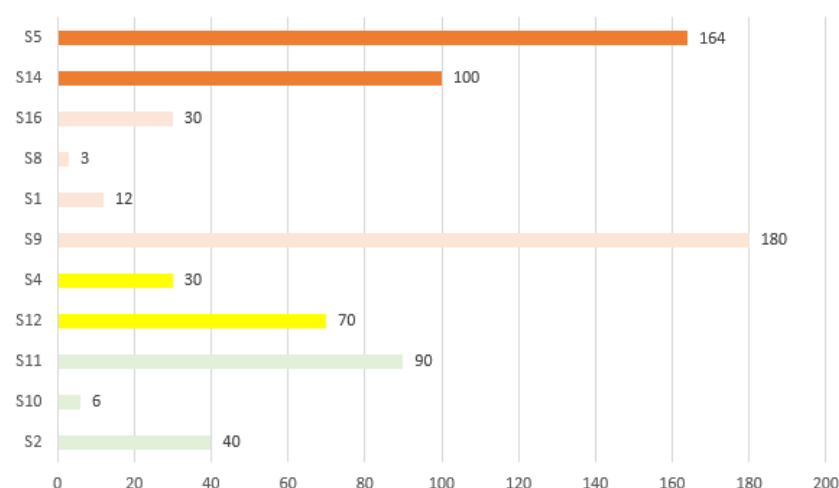
Quantity of Validation Events

For each startup, the number of validation events conducted prior to the development of a full MVP was identified and counted based on interview data and supporting startup materials. In this study, a validation event refers to a distinct activity aimed at testing assumptions related to the product, customer problem, market demand, or business model. Examples include customer interviews, surveys, prototype demonstrations, investor discussions, pilot projects, letters of

intent, and structured feedback sessions. The metric reflects the number of discrete validation activities performed, rather than the total number of participants involved in those activities. The number of users or stakeholders reached during these validation efforts was analyzed separately.

The study shows that all 16 startups engaged with potential users, customers, investors, and other relevant stakeholders to assess interest in their idea. A commonly held assumption is that increasing the amount of early-stage validation activity or user exposure may contribute to better-informed product decisions and potentially improve startup outcomes. However, within the studied cases, no clear relationship was observed between the number of validation-related interactions and subsequent startup performance. As illustrated in Figure 3, several lower-performing or discontinued startups reported relatively high levels of user and customer exposure during early validation activities. For example, S14, S5, and S9 reported respectively 100, 164, and 180 potential users or customers involved in validation-related activities, while two of these startups were no longer operational at the time of analysis. Within this sample, these findings suggest that the quantity of validation activities alone does not appear to explain startup performance outcomes — pointing instead toward the type and level of evidence those activities yield.

Figure 3: Number of validation events



Having established that the number of validation events does not predict startup performance, the analysis turns to a different question: what types of validation activities do startups engage in, and does the level of evidence those activities produce distinguish high-performing from low-performing ventures?

Three Stages of Validation

The literature on startup development identifies three broad stages that early-stage ventures typically move through before achieving commercial viability [111, 118, 31]. These stages provide the conceptual scaffold for the analysis that follows.

Problem validation

The initial stage of this framework emphasizes generating user value and considering informal stakeholder preferences. Most startups focus their efforts here, engaging in validation events through discussions with stakeholders and prioritizing the creation of user value. The primary

objective at this stage is to validate the fundamental interest in the idea. However, relying solely on this level of validation to build a startup is highly risky due to the inadequate level of validation it provides.

Problem-solution fit

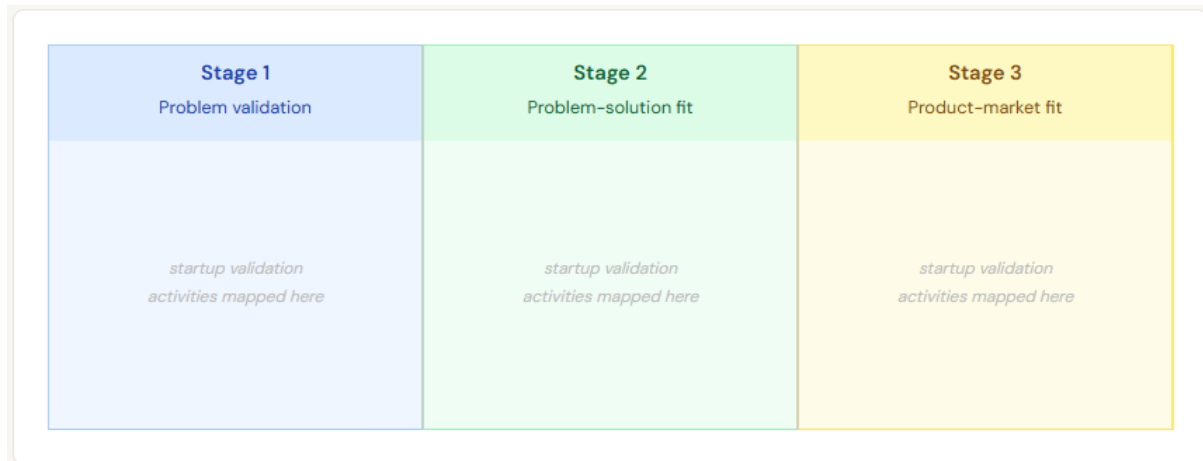
The second stage of the framework involves measuring the initial objective indicators of user interest in the product. This is achieved by counting explicit stakeholder requests for the idea and, more importantly, obtaining sign-ups from individuals willing to start using the product (for free) upon its release. At this stage, confidence in the venture idea increases, and the associated risk decreases. Many bootstrapped startups that decide to proceed with growing their venture have achieved this level of validation.

Product-market fit

Product-market fit, as operationalised in this study, refers to the point at which a startup demonstrates early commercial viability through external financial market commitment before the full product exists. This definition is intentionally more conservative than broader interpretations in the literature that rely primarily on interest signals, user growth, or free sign-ups, and reflects the survival-oriented perspective expressed by the founders interviewed. Within the Hierarchy of Validation, this stage is subdivided into two evidence levels: Level 5 (contingent investment commitments secured pre-MVP) and Level 6 (paying customers secured pre-MVP). While both levels represent strong forms of external validation, paying customers are positioned at Level 6 — the strongest evidence level observed in this framework — because direct willingness-to-pay from a customer receiving the intended value proposition constitutes a more immediate and product-specific market signal than investor commitment, which may additionally reflect confidence in the founders, vision, network, or broader market opportunity. In this sense, Level 6 represents the strongest form of autonomous market validation within the framework, as startups reaching this level may theoretically no longer require external funding to justify or sustain initial development. Within this sample, startups that secured Level 5 or Level 6 evidence prior to full MVP development were more frequently associated with stronger performance outcomes than startups relying primarily on lower-level validation signals. However, given the exploratory nature of the study, limited sample size, and contextual specificity of the cases, these observations should be interpreted as indicative associations rather than confirmed causal relationships and require further investigation in larger and more diverse startup populations.

These three stages provide the initial structure. Figure 4 presents the basic three-column Validation Canvas, with the three stages as columns. Each startup in the sample is assessed against these stages in the sections below.

Figure 4: Basic Validation Canvas – Three Stages drawn from literature



Coding Validation Activities: From Three Stages to Six Levels

To populate the Validation Canvas, all validation activities described by founders across the 29 interviews were extracted verbatim and coded inductively. Each activity was assigned a first-order code describing what the founder did, and a second-order code classifying the type of evidence produced. Table 20 presents the full coding results.

During the coding process, a consistent pattern emerged that motivated the subdivision of each of the three validation stages into two distinct levels. Within Problem Validation, a clear distinction exists between activities based primarily on informal personal experience, intuition, or exploratory network discussions, and activities grounded in more structured external research such as formal customer interviews, surveys, or systematic market exploration. While both contribute to reducing uncertainty, the latter provides stronger evidence because it relies less on founder assumptions and incorporates broader external input.

Within Problem-Solution Fit, a similar distinction emerged between validation activities that expose potential users to a concept without requiring commitment — such as prototypes, wireframes, landing pages, or Minimal Viable Experiments (MVEs) — and activities that involve observable behavioural commitment, such as free sign-ups, waiting lists, pilot agreements, or letters of intent. The latter were interpreted as stronger forms of evidence because they require an external party to invest time, attention, or reputational commitment beyond verbal interest alone.

Within Product-Market Fit, the subdivision follows the same underlying logic of increasing external commitment and decreasing uncertainty. Securing investment commitments before MVP development (Level 5) requires an external party to commit financial resources based on confidence in the founders, team, vision, or perceived market opportunity. Securing paying customers before MVP development (Level 6), however, requires an external party to directly exchange money for the proposed value proposition itself, thereby validating willingness-to-pay more directly and independently from investor judgement. For this reason, paying customers were interpreted as the strongest form of pre-MVP market evidence observed within the studied cases. A startup reaching Level 6 has demonstrated that at least one external stakeholder is willing to financially commit to the proposed value before the complete product

exists. Together, these distinctions resulted in a six-level hierarchy of progressively stronger evidence and decreasing uncertainty.

Table 22. Coding results for Validation Canvas, Validation activities coded per startup

S#	Level	Stage	Activity type (1st-order code)	Supporting quote
S1	L1	Problem Validation	Informal discussions	"Talked with 12 clubs, one of the founders saw the struggle at his club and started thinking about how to solve it." — S1
S7	L1	Problem Validation	Own experience as validation	"Vision trumps cost. We only release when it's perfect — we just consider the whole market and think about the percentage we could capture." — S7
S8	L1	Problem Validation	Own experience as validation	"Pure own itch — ongoing through very close interaction of the other company of Kevin. No external validation upfront." — S8
S14	L1	Problem Validation	Informal company discussions	"We started by asking 25 to 30 companies if it was a good idea, started to create wireframes." — S14
S17	L1	Problem Validation	Internal co-founder discussion	"Showed this POC to the 2nd initial co-founder, and he found it interesting — was selling trends research but without actual data backing." — S17
S2	L2	Problem Validation	Customer interviews (30-40)	"Done 30–40 customer interviews with CTOs about their security problems, how they solve it now ... His reputation does help, already has 900k on possible rev based on the discussions." — S2
S4	L2	Problem Validation	Battle-tested with 30-35 profiles	"Ideation started 2 years ago where the idea got form through brainstorm, drawings, initial discussions until a decent business model got found — this got battle tested with 30 to 35 different profiles." — S4
S5	L2	Problem Validation	Survey (164 customers)	"Before getting started, we conducted a survey with 164 customers. The answers from this survey have been the guiding red line for our product strategy and vision." — S5
S7	L2	Problem Validation	Surveys and interviews (~350)	"A year before getting officially started they had been sending out multiple surveys and had many interviews: around 350 people in total about the product idea." — S7
S9	L2	Problem Validation	150 meetings across 4 industries	"Having had 150 meetings across 4 industries to validate the idea... over 30 design agencies to further increase feedback and insights." — S9

S11	L2	Problem Validation	Survey (30-40 freelancers + 50 doctors)	"Checked with 30 or 40 freelancers via mail, and got response via a survey of 50 doctors that they were positive about the idea." — S11
S13	L2	Problem Validation	Google forms validation	"Sent out some Google forms for validation on customers and influencers. Built a MVP with an external partner (without a paywall)." — S13
S16	L2	Problem Validation	Survey (300) + 20-30 B2B interviews	"300 filled out the survey — needed to create and train the data model. Talked with 20 to 30 companies to validate the idea." — S16
S1	L3	Problem-Solution Fit	Free MVP shared with 12 clubs	"Shared the FREE MVP with 12 clubs. Feedback loop: clubs → get feedback (call them) & continue." — S1
S2	L3	Problem-Solution Fit	Wireframes used to sell product	"Based on created wireframes, we started trying to sell the product. This stealth mode approach allowed us to validate our concept before writing a single line of code." — S2
S4	L3	Problem-Solution Fit	MVP built + hackathons as sandbox	"They built in the product track a MVP. Did some hackathons as kind of sandbox plays (and win some money)." — S4
S9	L3	Problem-Solution Fit	High-fidelity prototype	"Build a prototype — got feedback again (needed a bit more to get started); spin off (of VUB)." — S9
S12	L3	Problem-Solution Fit	Working prototype + 2 years prep	"Had already a working prototype to sell. Had already been in contact with the government and many of the linked institutions. All of this has taken around 2 years of preparations." — S12
S13	L3	Problem-Solution Fit	MVP with external partner	"Build a MVP with an external partner (without a paywall). Started with an app." — S13
S14	L3	Problem-Solution Fit	Wireframes + development started	"Got 500.000 imec funding — pure based on vision, found xxx as new dev capacity; started with writing specs and the development of the first version." — S14
S15	L3	Problem-Solution Fit	POC used for pitching	"Showed this POC to the 2nd initial co-founder. They decided to improve the POC and make it company aware. This improved POC got used to start pitching, including imec istart." — S15
S16	L3	Problem-Solution Fit	Test case Shopify MVP	"Created a test case Shopify MVP (with her sister) to see the B2C results." — S16
S17	L3	Problem-Solution Fit	POC improved and used to pitch	"From there they signed 1 (big) customer, and decided to start building it for them." — S17

S18	L3	Problem-Solution Fit	Simple lead generation website	"The moment they started looking for validation they created a page via Notion, and after that a very simple lead generation website." — S18
S9	L4	Problem-Solution Fit	Reflection: needed paid commitments earlier	"Sell the idea faster, talk to more potential customers to actually sell the product if only with a prototype, so that we would have more clearly defined features and paid commitments at hand." — S9
S10	L4	Problem-Solution Fit	1 customer signed pre-seed	"In November went to Brazil to talk with 6 potential customers, of which 1 signed on, as well as a pre-seed investor that agreed." — S10
S12	L4	Problem-Solution Fit	70/450 institutions signed up	"Signed up 70/450 institutions when taking over the site — mainly in West-Flanders. All of this has taken around 2 years of preparations." — S12
S14	L4	Problem-Solution Fit	3000 candidates in database	"We're smooth on the candidate side, with over 3000 in our database." — S14
S16	L4	Problem-Solution Fit	2 non-paying customers found	"Found 2 possible (non-paying) customers." — S16
S2	L5	Product-Market Fit	2M raised (first round) at 100 paying customers	"2M first round: 500k subsidies, 500k founders, 1M angels. Went to raise again only at 100 paying customers — VC wanted in." — S2
S10	L5	Product-Market Fit	Pre-seed investor secured alongside first customer	"Having everything lined up from customers, investment and MVP, got started." — S10
S14	L5	Product-Market Fit	500k imec funding (vision-based)	"Got 500.000 imec funding — pure based on vision. Working towards a 1M funding round till the end of the year. [Still went bankrupt]" — S14
S2	L6	Product-Market Fit	Sold 6x before writing a line of code	"xxx: been preparing over 1 year before getting started, also with interviews — had already sold it 6 times without having written a single line of code — stealth mode. 900k on possible rev based on the discussions." — S2
S4	L6	Product-Market Fit	1 customer signed in Brazil trip	"In November went to Brazil to talk with 6 potential customers, of which 1 signed on." — S4
S10	L6	Product-Market Fit	First customer secured pre-seed	"Talked with potential customers and secured our first customer as well as a pre-seed investor." — S10
S12	L6	Product-Market Fit	30k recurring revenue, 100% margin	"They do have revenue by working as an agency ... There are three packages: free, paid, premium, and they currently already

				have a 30.000 in recurring rev at 100% margin." — S12
S14	L6	Product-Market Fit	100k signed, 180k prospects	"For customers, we have 100k signed and also have 180k in prospects." — S14
S15	L6	Product-Market Fit	Ageas signed as first customer	"From there they signed 1 (big) customer: xxx." — S15
S17	L6	Product-Market Fit	Pepsico signed 120k	"Did sign xxx: 120k." — S17
S18	L6	Product-Market Fit	20-25k revenue before formally starting	"By 05/2020, they had already generated 20 to 25k in revenue and decided to really get started." — S18

L5 = Funding pre-MVP (contingent investment). L6 = Paying customers pre-MVP (strongest evidence — direct willingness-to-pay). S14 entries marked [went bankrupt despite this] serve as the necessary-but-not-sufficient exception.

Figure 5 reveals that all 16 startups conducted some form of validation activity, and most conducted activities at multiple levels. The six levels now constitute the Validation Canvas — a mapping instrument that plots where each startup's evidence fell before committing to full MVP development.

Figure 5 presents the updated Validation Canvas with all six levels and all 16 startups mapped. Each startup may appear in multiple cells, reflecting the diversity of activities across more than one level.

With all activities mapped, the Canvas constitutes what we term the Validation Canvas. The next step is to determine whether patterns on this Canvas are associated with venture performance.

Performance Patterns on the Validation Canvas

To examine whether the Canvas reveals performance-related patterns, the startups are separated by performance tier and mapped individually. The observations below are neutral and descriptive

The three Tier 1 startups (see Figure 6) — S2, S10, and S11 — each show a different profile on the Canvas, but share one common characteristic: all three have Level 6 activity before their MVP development commenced.

"Done 30–40 customer interviews with CTOs about their security problems ... already has 900k on possible revenue based on the discussions." — S2, founder (Tier 1)

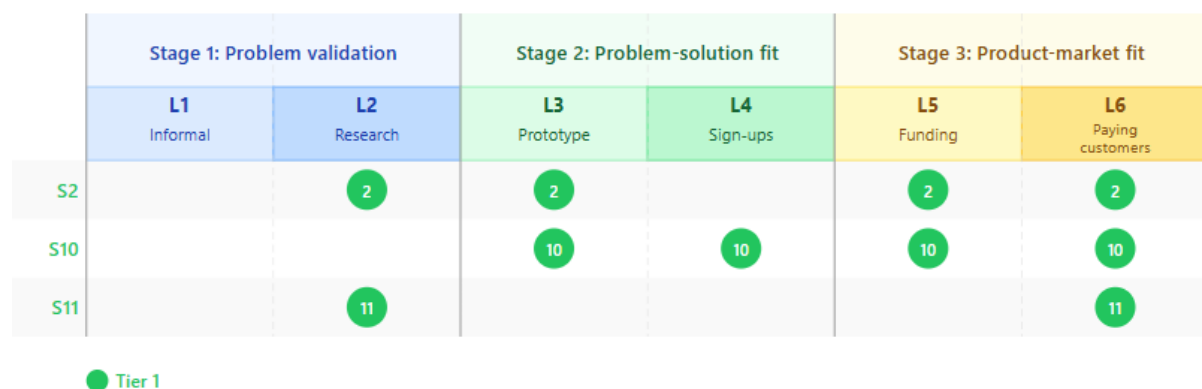
"Based on created wireframes, we started trying to sell the product. We tried selling it first through wireframes before writing any code and having signed potential revenues of around [xxx] thousand euros and funding." — S2, founder (Tier 1)

"Having everything lined up from customers, investment and MVP, before getting started." — S10, co-founder (Tier 1)

Figure 5: Updated Validation Canvas (3x2) – Three Stages, Six Evidence Levels



Figure 6: Tier 1 – Top Performers



Tier 1 startups. Note concentration at L5–L6.

S11 presents one of the most instructive Tier 1 cases within the Validation Canvas. Their Canvas shows strong Level 2 evidence through structured surveys with freelancers and doctors, but no observable Level 3, Level 4, or Level 5 validation activity prior to MVP development. In practice, they moved directly from Problem Validation into development without first progressing through prototype-based validation, behavioural commitment signals, or pre-MVP investment. According to the founders interviewed, this decision nearly caused the venture to fail. Although the startup ultimately reached Level 6 through paying customers and generated revenue, this occurred under significant financial runway pressure rather than as part of a deliberate and structured validation strategy:

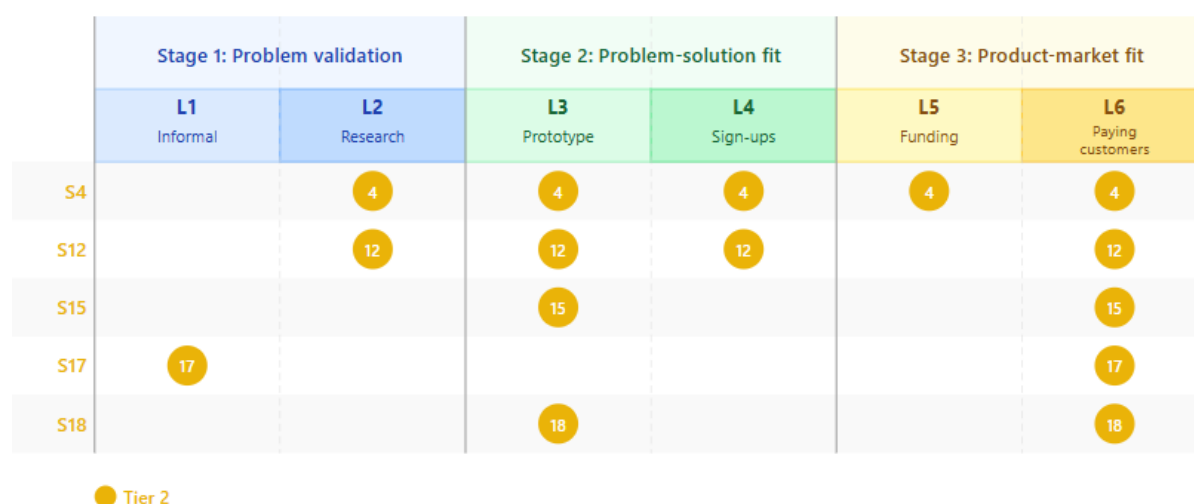
"We checked with 30 or 40 freelancers ... and conducted a survey with 50 doctors to gauge their interest... The feedback was encouraging, so we decided to move forward." — S11, CEO (Tier 1)

"... the latest feature we prioritised could generate revenue, which was crucial as our runway was almost finished." — S11, CEO

"I would have started with a technical co-founder or hired an in-house developer from the beginning." — S11, CEO — on what they would do differently

S11 is currently in Tier 1, but this was achieved despite the gap in their validation trajectory, not because of it. Their Canvas profile is an outlier within the tier.

Figure 7: Tier 2 – Mid-high Performers



Tier 2 startups. Note divers validation activities but high concentration at L6.

All five Tier 2 (see Figure 7) startups reached Level 6 (paying customers) before or at the point of committing to full MVP development. S4 is the only Tier 2 startup that also reached Level 5 (pre-MVP funding), alongside S2 and S10 in Tier 1. This means that funding pre-MVP (L5) was secured by only three startups in the entire sample — all in Tiers 1 or 2 — while paying customers pre-MVP (L6) was reached by all eight Tier 1 and Tier 2 startups, plus S14 in the OOB group.

S18 is notable for the simplicity and speed of their approach — a basic lead generation website generated immediate revenue before any structured development began:

"The moment they started looking for validation they created a page via Notion, and after that a very simple lead generation website. By 05/2020, they had already generated 20 to 25k in revenue and decided to really get started." — S18, co-founder (Tier 2)

S12 reached L6 through a different route — by leveraging a government website takeover to acquire 70 institutions as implicit users, then building a revenue model on top:

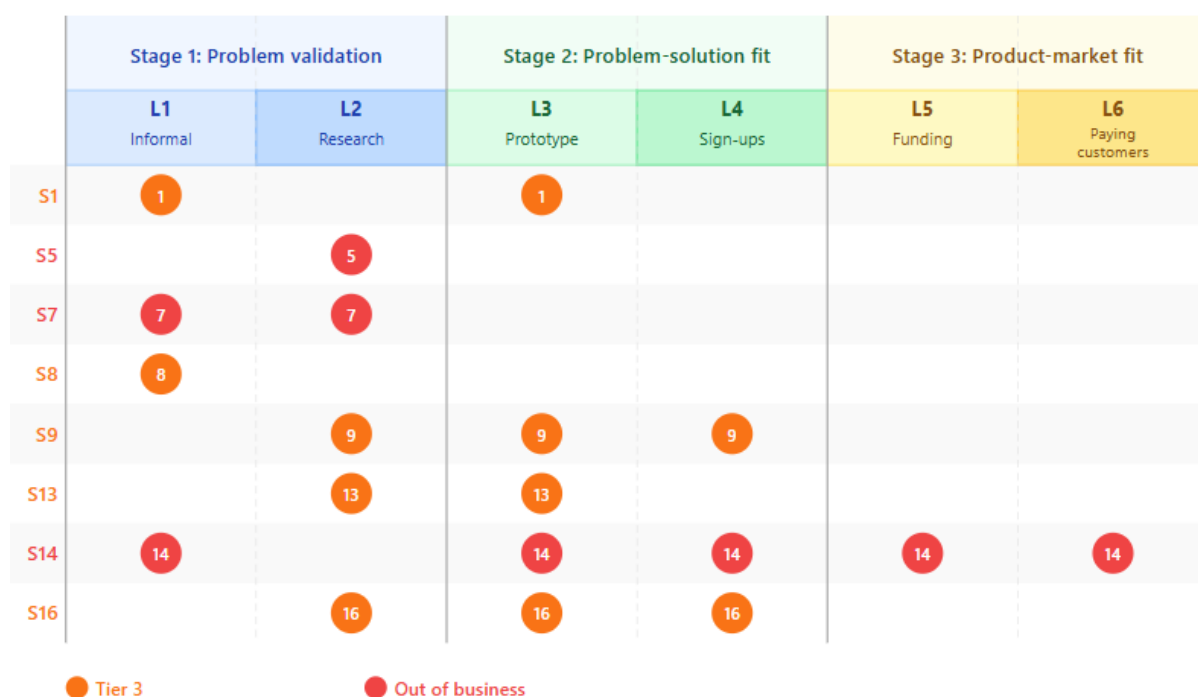
"Signed up 70/450 institutions when taking over the site — mainly in West-Flanders. There are three packages: free, paid, premium, and they currently already have a 30.000 in recurring rev at 100% margin." — S12, founder (Tier 2)

S15 converted a POC demonstration directly into a signed contract with a big customer before any formal MVP existed:

"From there they signed 1 (big) customer: xxx. Decided to start building it for xxx, didn't invoice yet. The amount got too high, which forced them to invoice and started the company." — S15, co-founder (Tier 2)

S4 completed the most comprehensive validation trajectory of any startup in the tier — hackathons, customer meetings in Brazil, a signed customer, and a pre-seed investor — all before MVP development. Their Canvas profile closely resembles S2 and S10.

Figure 8: Tier 3 and Out-of-Business



Tier 3 and Out-of-business startups. Note limited, to no concentration at L5-L6

The pattern observed for Tier 3 and out-of-business startups (see Figure 8) further reinforces the distribution of evidence levels within the Validation Canvas. No Tier 3 startup secured

Level 5 evidence (pre-MVP funding commitments) or Level 6 evidence (pre-MVP paying customers) before committing to development. Among the out-of-business startups, S14 represents the notable exception, having reached both Level 5 and Level 6 prior to MVP development, yet still ultimately failing due to execution-related challenges discussed earlier. This exception illustrates that while higher levels of market evidence may be associated with stronger startup performance within this sample, they do not guarantee long-term success.

The out-of-business startups S5 and S7 both show strong L2 activity — extensive surveys and interviews — followed by a direct jump to development with no L3–L6 evidence. S5's CEO described their survey as a permanent reference point:

"Before getting started, we conducted a survey with 164 customers. The answers from this survey have been the guiding red line for our product strategy and vision." — S5, CEO (out of business)

S7 went even further in terms of raw validation volume — approximately 350 people surveyed and interviewed — yet still proceeded without financial commitment from the market:

"A year before getting officially started they had been sending out multiple surveys and had many interviews: around 350 people in total about the product idea." — S7 (out of business)

S8 shows the sparsest Canvas profile of all startups — a single L1 activity (own-itch experience) before building. No external validation was conducted:

"Pure own itch — ongoing through very close interaction of the other company of Kevin. No external validation upfront." — S8, co-founder (Tier 3)

S9 shows an extensive L2 and L3 profile — 150 meetings, design agency feedback, and a prototype — but was unable to convert these activities into L5 evidence. The co-founder later identified this as the key thing they would do differently:

"Sell the idea faster, talk to more potential customers to actually sell the product if only with a prototype, so that we would have more clearly defined features and paid commitments at hand." — S9, co-founder (Tier 3) — on what they would do differently

S14 is the critical exception in this group — the only out-of-business startup that reached L5 and L6 before MVP development. Their Canvas shows activity at every level, with 100k in signed customer contracts and 500k imec funding secured. Yet the venture filed for bankruptcy. The CEO's account traces a textbook progression:

"We started by asking 25 to 30 companies; they thought it was a good idea and created some wireframes." — S14, CEO (OOB) — on Problem Validation

"We're smooth on the candidate side, with over 3000 in our database." — S14, CEO — on Problem-Solution Fit

"For customers, we have 100k signed and also have 180k in prospects." — S14, CEO — on Product-Market Fit

S14's profile is important for interpreting the overall pattern, as it illustrates that reaching higher levels of the Validation Canvas — including Level 5 and Level 6 evidence — may be associated with stronger startup performance, but does not guarantee success. Despite securing substantial market evidence prior to development, including funding and paying customers, co-founder conflicts, developer departures, and accumulated operational debt ultimately prevented successful execution. This case reinforces the exploratory nature of the framework: the Validation Canvas describes an observed pattern within this sample rather than a deterministic or sufficient condition for startup success.

Comparative Observation: Evidence Level and Performance

Comparing the three Canvas maps, a distributional pattern emerges within this sample. Top-performing and mid-performing startups show a concentration of evidence at Level 6 — and, among the strongest performers, Level 5 — before or around the point of substantial MVP development. Lower-performing and out-of-business startups concentrate primarily at Levels 1 and 2, with the notable exception of S14.

More precisely, all Tier 1 and Tier 2 startups ultimately reached Level 6 evidence, meaning that they secured paying customers before or around the point at which substantial development commitment occurred. None of the Tier 3 startups reached Level 6 prior to failure or stagnation. S14 is the only out-of-business startup that reached both Level 5 and Level 6 evidence before ultimately failing due to execution-related challenges. Level 5 evidence was observed only among three startups (S2, S4, S10), suggesting that pre-MVP investment commitments may represent an additional marker of the strongest-performing ventures rather than a prerequisite for stronger performance more generally.

The central observation is processual rather than tautological. The finding is not simply that stronger-performing startups eventually obtained customers or funding. Rather, within this sample, startups that secured stronger forms of market commitment — particularly paying customer commitment — before or during the transition toward substantial MVP development were more frequently associated with stronger performance outcomes than startups relying primarily on lower-level validation signals such as interviews, surveys, or informal expressions of interest. This points toward the timing and strength of market evidence at the moment substantial development commitment occurs as a potentially important differentiating factor. Given the exploratory nature of the study, limited sample size, and contextual specificity of the cases, these findings should be interpreted as indicative associations rather than confirmed causal relationships and require further investigation in larger and more diverse startup populations.

An Exploratory Note on Serial Founder Experience

An exploratory observation that warrants future investigation concerns founder experience. Of the three founders in the sample who could be classified as serial entrepreneurs — S2 (previously co-founder of 2 successful scale ups), S4 (multiple prior exits), and S7 (significant prior cash-out via Microsoft acquisition) — two (S2 and S4) reached the highest evidence levels first — both paying customers (L6) and funding (L5) — before conducting broad exploratory validation, consistent with a top-down traversal. S7 did not, describing a philosophy that explicitly deprioritised financial validation before building:

"Vision trumps cost. ROI calculation? No, not really, just consider the whole market and think about the percentage that they could capture." — S7, co-founder (serial, OOB)

With only three cases that do not agree with each other, no pattern can be claimed. This observation is flagged solely as a proposition for future research: founder experience may moderate the direction in which the validation hierarchy is traversed.

From Canvas to Hierarchy: The Hierarchy of validation

Building on the Validation Canvas analysis and the comparative patterns observed across the studied cases, the six identified levels appear to function as a hierarchy of evidence within this sample, in the sense that higher levels of market commitment prior to substantial MVP development were more frequently associated with stronger startup performance outcomes. This motivates the naming of the framework as the **Hierarchy of Validation**: a staged and bidirectional evidence framework in which validation activities are ordered according to increasing levels of external commitment and decreasing levels of uncertainty. The framework is grounded in the empirical patterns observed within the current sample and is proposed as an exploratory model for future investigation rather than a validated law or deterministic success model.

The ordering of the hierarchy follows a consistent underlying logic: each level requires progressively greater external effort, behavioural commitment, reputational exposure, or financial risk from an outside party. At the lower levels, startups primarily relied on informal discussions, founder intuition, exploratory stakeholder conversations, surveys, or interviews. While these activities may provide useful directional insights, they generate relatively weak forms of evidence because they require limited commitment from external stakeholders. As startups move upward through the hierarchy, the observed validation activities increasingly require observable market commitment. Prototypes, sign-ups, funding commitments, and paying customers therefore represent progressively stronger forms of validation because they involve increasing levels of behavioural or financial commitment under uncertainty.

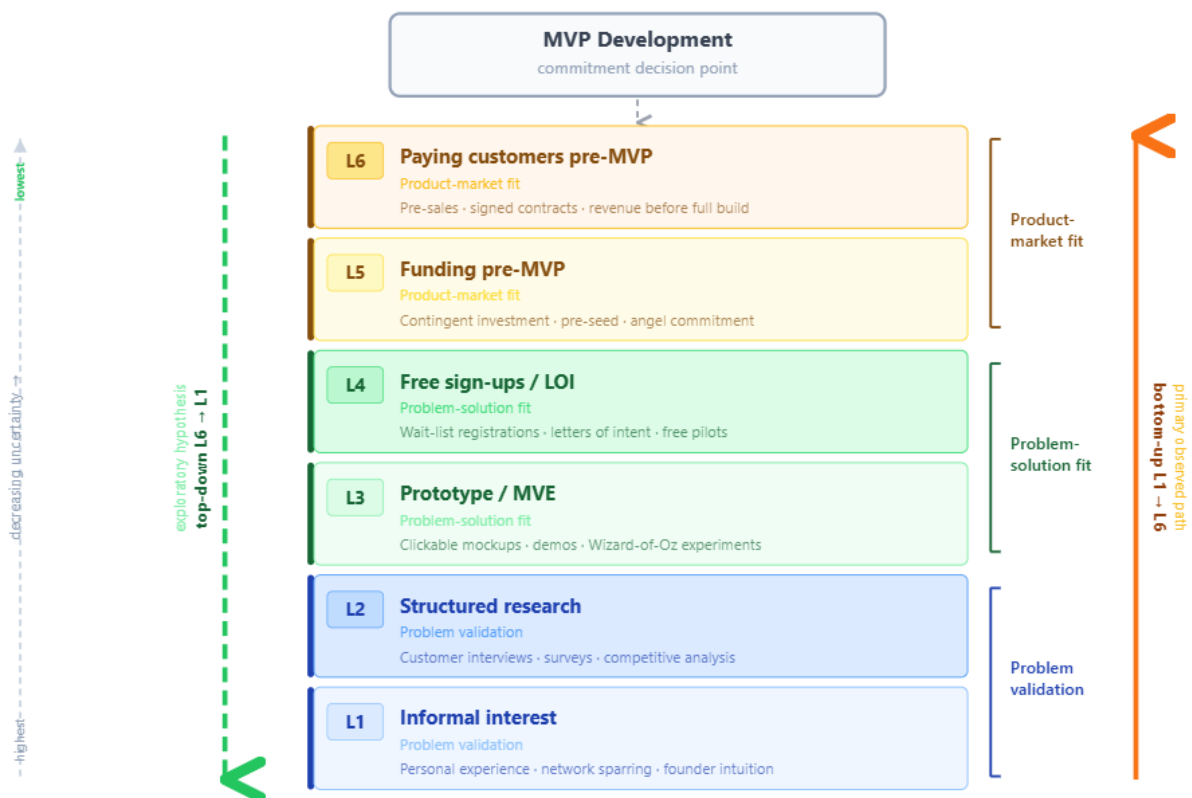
Within Product-Market Fit, the distinction between Level 5 and Level 6 emerged as particularly important. Contingent investment commitments secured pre-MVP (L5) represent strong evidence because external parties commit financial resources based on confidence in the founders, vision, or perceived market opportunity. Paying customers secured pre-MVP (L6), however, represent the strongest form of evidence observed within this framework because they involve direct willingness-to-pay for the proposed value proposition itself, without reliance on intermediary investor judgement. In this sense, a startup reaching Level 6 may theoretically no longer require Level 5 validation, as the market has already demonstrated direct financial commitment to the proposed value.

The connection between the hierarchy and startup performance is processual rather than tautological. The observation is not simply that stronger-performing startups eventually obtain customers or funding, but that the process of securing stronger forms of market commitment before committing substantial development resources appears associated with stronger outcomes within this sample. In particular, securing paying customer commitment prior to or during the transition toward substantial MVP development emerged as the most consistent differentiating pattern across the higher-performing startups. The hierarchy

therefore provides a structured way of reasoning about when sufficient market evidence may exist to justify substantial development investment.

Across the studied cases, the primary direction observed was bottom-up traversal through the hierarchy, moving from informal interest signals (L1) toward direct financial market commitment (L6) as evidence strength increased. However, the findings also suggest a potential bidirectional interpretation. While most founders appeared to progress sequentially upward through the hierarchy, some cases indicate that experienced founders may leverage existing reputation, networks, or prior entrepreneurial credibility to secure higher-level commitments earlier in the process, effectively traversing the hierarchy top-down. In such situations, lower-level validation activities may serve more as confirmation than discovery. This top-down interpretation remains exploratory and is proposed as a hypothesis for future research. Figure 9 presents the resulting Hierarchy of Validation and its six evidence levels.

Figure 9: The Hierarchy of Validation as a bidirectional process model.



L1 (weakest evidence) to L6 — Paying customers pre-MVP (strongest evidence).

Bottom-up path (orange, L1→L6): primary direction observed.

Top-down path (green, dashed, L6→L1): exploratory hypothesis for future research

4.5. Discussion

The Hierarchy of Validation as a Directional Evidence Framework

The Hierarchy of Validation is inspired by, but not based on, Maslow's Hierarchy of Needs [142] — a model that has been subject to substantial empirical and theoretical critique [143, 144]. The pyramid metaphor is therefore intentionally not adopted. Instead, the framework is

presented as a directional process model that organizes validation activities according to increasing levels of external market commitment and decreasing levels of uncertainty, rather than as a fixed stage theory.

Importantly, the hierarchy does not imply that startups must progress linearly through every level, nor that reaching higher levels guarantees success — S14 illustrates this clearly. Rather, the framework provides a structured lens through which founders and researchers can reason about what level of market evidence may be sufficient to justify committing substantial resources to full MVP development. Within this sample, lower levels of validation such as informal discussions, surveys, exploratory interviews, or general expressions of interest (L1–L2) did not appear sufficient on their own to distinguish stronger-performing startups from lower-performing ones. Instead, stronger-performing startups were more frequently associated with higher levels of behavioural or financial commitment prior to full development, particularly paying customers or contingent investment commitments (L5–L6).

This observation should not be interpreted as a deterministic gating mechanism or universal threshold. Rather, it represents an exploratory association observed across the current cases. Future research using larger and more diverse samples is required to examine whether this apparent threshold generalises across startup contexts.

The bidirectional nature of the hierarchy captures an additional observation emerging from the cases. While most founders appeared to traverse the hierarchy bottom-up — progressively accumulating stronger forms of evidence — some cases suggest that experienced founders may be able to leverage existing reputation, networks, or prior entrepreneurial credibility to secure higher levels of commitment earlier in the process. In such situations, lower-level validation activities may serve more as confirmation than discovery. This directional interpretation remains exploratory and should be examined further in future research.

The Hierarchy as a Decision Framework

Beyond its descriptive contribution, the Hierarchy of Validation also has implications as a decision framework. The central question emerging from the findings is when a startup has validated enough to justify committing substantial resources to MVP development.

Within this sample, the observed pattern tentatively suggests that the strongest differentiator between stronger- and lower-performing startups was not the quantity of validation activities conducted, but whether startups secured stronger forms of external market commitment prior to substantial development investment. In particular, startups relying primarily on Level 1 or Level 2 evidence — informal interest signals, exploratory discussions, surveys, or structured research — were more frequently associated with weaker performance outcomes when substantial MVP development was initiated before stronger forms of behavioural or financial commitment had been secured.

Lower-level validation activities remain important because they help founders explore problems, understand stakeholder needs, and reduce initial uncertainty. However, these activities primarily capture stated intentions or perceived interest rather than observable market behaviour involving meaningful commitment or financial risk. Within this sample, stronger-performing startups were more frequently associated with higher levels of evidence involving direct behavioural or financial commitment from external stakeholders.

More specifically, Level 6 evidence — securing at least one paying customer prior to or during the transition toward substantial MVP development — emerged as the strongest and most consistent form of pre-MVP validation observed across the higher-performing startups. Unlike lower-level validation signals such as interviews, surveys, sign-ups, or expressions of interest, Level 6 involves direct willingness-to-pay from an external stakeholder with their own financial resources at risk. This represents a qualitatively different form of evidence because it validates the proposed value proposition through actual market behaviour rather than stated intention alone.

Level 5 evidence — contingent investment commitments secured pre-MVP — also appeared associated with stronger-performing startups, particularly among the highest-performing ventures. However, investment commitments differ conceptually from Level 6 evidence in that they may additionally reflect confidence in the founders, team, network, or market opportunity rather than direct validation of the product itself. Within this sample, several startups reached strong performance outcomes without pre-MVP investment, whereas no Tier 3 startup secured Level 6 evidence prior to substantial development commitment.

Importantly, these findings should not be interpreted as deterministic decision rules or universal thresholds. Rather, they represent exploratory associations grounded in the observed cases and should be interpreted accordingly. Nevertheless, the framework suggests that validation may be more usefully interpreted as a threshold of evidence strength to be reached before committing substantial development resources, rather than merely as a sequence of validation activities to complete.

Relationship to Lean Startup and Continuous Experimentation

The Hierarchy of Validation is conceptually aligned with the Lean Startup methodology [124] and the hypothesis-driven Customer Development approach [137, 138]. All three perspectives emphasize reducing uncertainty through iterative validation before committing significant resources to full-scale development. The observed pattern in this study — namely that stronger-performing startups more frequently secured stronger forms of market commitment prior to building — is consistent with the broader build-measure-learn logic underpinning these approaches.

However, the contribution of the Hierarchy of Validation differs in focus. Rather than prescribing iterative experimentation cycles, the framework organizes validation activities according to the strength of evidence and external commitment they represent. In doing so, it provides a more granular process-oriented lens for reasoning about when validation may become sufficiently convincing to justify progression toward MVP development.

The framework also differs from Continuous Experimentation (CE). CE focuses on improving business models and already-built software products through iterative experimentation using behavioural user data [115]. As such, CE presupposes the existence of a software product and the technical infrastructure required to support experimentation at scale [115, 117]. The Hierarchy of Validation operates earlier in the startup lifecycle, before substantial software development has occurred. Its primary purpose is to structure pre-build validation activities aimed at reducing uncertainty and establishing sufficient market evidence to justify investment into MVP development. The distinction between both approaches is therefore primarily temporal rather than philosophical: the Hierarchy focuses on pre-build validation, whereas CE becomes applicable post-build.

Limitations and Future Research

This study is explicitly exploratory in nature. The findings are generated from 16 software startup cases originating from a single Belgian accelerator cohort, which imposes important limitations on the generalisability of the observed patterns. The associations identified between validation evidence levels and startup performance should therefore not be interpreted as confirmed causal relationships, but rather as empirically grounded observations within a small and contextually homogeneous sample. In addition, the study relies primarily on retrospective founder interviews, introducing potential risks related to recall bias and post-hoc rationalisation.

Several additional limitations should also be acknowledged. While the study identifies differences in the level of evidence reached by startups prior to substantial MVP development, it does not systematically measure the quality of the underlying validation activities themselves. As a result, conclusions cannot be drawn regarding whether certain interviews, surveys, experiments, prototypes, or customer interactions were conducted more rigorously or effectively than others. Similarly, startup performance is inherently multidimensional and dynamic, meaning that the tiering approach used in this study should be interpreted as an analytical simplification rather than an absolute measure of venture success.

Three directions for future research emerge from these limitations. First, the central association observed in this study — namely that startups securing stronger forms of market commitment prior to substantial development investment, particularly paying customer commitment (L6), were more frequently associated with stronger performance outcomes — requires confirmatory investigation using larger, more diverse, and ideally prospective longitudinal research designs. The distinct role of pre-MVP investment commitments (L5) as a potential marker of the strongest-performing ventures also warrants separate investigation. Second, the proposed bidirectional interpretation of the hierarchy, particularly the hypothesis that prior entrepreneurial experience may influence traversal direction through the levels, should be examined explicitly in future empirical studies. Third, the quality and reliability of validation activities themselves remain an important open research question. Future research should therefore investigate factors such as hypothesis specificity, customer representativeness, falsifiability, and the strength of behavioural or financial commitment generated through different forms of validation activity.

4.6. Conclusion

This study explored how early-stage software startups validate their initial idea before committing to the development of their first Minimum Viable Product (MVP), and whether differences in pre-MVP validation behaviour are associated with startup performance. Based on 29 semi-structured interviews with founders from 16 software startups, the analysis produced three exploratory findings.

First, within this sample, the quantity of validation activities conducted before MVP development does not appear to distinguish stronger-performing startups from lower-performing ones. Startups across all performance tiers engaged in interviews, surveys, discussions, prototypes, and experiments, with some lower-performing startups conducting as many — and in some cases more — validation activities than stronger-performing cases. These

findings suggest that validation quantity alone is not a meaningful differentiating factor in early-stage startup performance.

Second, the findings indicate that the level of evidence generated through validation activities appears more relevant than the number of activities performed. Startups that secured stronger forms of external market commitment prior to substantial MVP development — particularly paying customers (Level 6) and, to a lesser extent, contingent investment commitments (Level 5) — were more frequently associated with stronger performance outcomes within this sample. More specifically, all Tier 1 and Tier 2 startups ultimately reached Level 6 evidence before or around the point at which substantial development commitment occurred, whereas no Tier 3 startup did. Level 5 evidence was observed only among a smaller subset of the strongest-performing startups, suggesting that pre-MVP investment may function as an additional differentiator among top performers rather than as a prerequisite for stronger performance more generally. The contribution of this finding is processual rather than tautological. The observed distinction is not simply that stronger startups eventually obtained customers or funding, but that they appeared more likely to secure stronger forms of market commitment before committing substantial development resources. Product-market fit, as operationalised in this study, therefore refers to demonstrated commercial viability through financial market commitment before the complete product exists. At the same time, the findings also show that reaching these higher levels of validation does not guarantee success, as illustrated by S14, where execution-related challenges ultimately prevented sustainable growth despite strong market evidence.

Third, the study organizes the identified validation activities into the Hierarchy of Validation: a staged and bidirectional process model in which validation activities are structured according to increasing levels of external commitment and decreasing levels of uncertainty. Within this framework, Level 6 — paying customers secured prior to full MVP development — represents the strongest form of pre-MVP evidence observed in the studied cases, as it directly validates willingness-to-pay without relying on intermediary investor judgement. Across the cases, bottom-up traversal through the hierarchy, moving from informal interest signals toward direct financial commitment, emerged as the dominant pattern among stronger-performing startups. The framework therefore provides an exploratory lens for reasoning about when market evidence may become sufficiently convincing to justify committing substantial resources to development. Within this sample, lower-level validation signals such as interviews, surveys, or general expressions of interest alone did not appear sufficient to distinguish stronger-performing startups from lower-performing ones, whereas stronger behavioural and financial commitments appeared more closely associated with stronger outcomes. The proposed top-down traversal of the hierarchy, in which experienced founders potentially leverage reputation, networks, or prior entrepreneurial credibility to secure higher-level commitments earlier in the process, remains a hypothesis for future research.

The Hierarchy of Validation is inspired by, but not derived from, Maslow's Hierarchy of Needs and is intentionally presented as a directional process framework rather than a deterministic stage theory. Given the exploratory nature of the study, the limited sample size, and the contextual focus on startups originating from a single accelerator ecosystem, the findings should be interpreted as empirically grounded observations rather than generalisable causal claims. Further research involving larger, more diverse, and longitudinal samples is required to validate the proposed framework and further examine the relationship between evidence strength, validation sequencing, and startup performance.

5. What Really Matters in Requirements Prioritization: Evidence from 171 Practitioners in Software Startup and Mature Company Contexts

Abstract⁵

Requirements Prioritization (RP) is a cornerstone of software product management and requirements engineering, directly shaping which features are developed, deferred, or abandoned. Yet little is known about how prioritization criteria vary between organizational contexts such as startups and mature companies. To address this gap, we conducted a large-scale survey of 171 practitioners from 37 countries and 51 industries, including 36.8% who had founded at least one startup and an average of six years of product management experience. Respondents were randomly assigned to consider either a startup or mature company context when answering structured and open-ended questions on prioritization categories and criteria. The results reveal clear context-specific patterns. Startups place over twice as much emphasis on expected cost compared to mature companies (context score 2.15), and also prioritize time-to-market and minimum viable product (MVP) requirements coverage. Mature companies, by contrast, assign greater weight to revenue generation, return on investment (ROI), and quality metrics. Customer value was consistently ranked highly in both contexts (>30% of responses), though startups tended to rely on informal validation, while mature firms incorporated structured financial and quality assessments. This study provides one of the largest empirical investigations of RP practices to date. It contributes by adding to the requirements engineering literature — where cost-efficiency as a dominant startup prioritisation driver remains largely absent — evidence that aligns with broader insights from the entrepreneurship literature on resource scarcity and effectuation, and by offering evidence-based insights for tailoring context-sensitive prioritization methods.

Keywords

Requirement Prioritization, Requirements Engineering, Product Management, Software Startups, Mature Companies, Prioritization Criteria, Time to Market

5.1. Introduction

Requirements Prioritization (RP) lies at the heart of software development decision-making, determining which features to build, defer, or abandon. The choice of prioritization criteria directly shapes product roadmaps, financial outcomes, and ultimately the survival of ventures. Prior research has identified a wide range of requirements prioritization criteria (RPC)—including cost, revenue, risk, time, customer value, and technical feasibility—but the relative importance of these criteria has often been studied in isolation, without explicit attention to organizational context [53, 145, 146].

Startups and mature companies face markedly different conditions. Startups operate under severe funding constraints, short runways, and high uncertainty, where a single failed prioritization decision can lead to insolvency [24, 27, 28]. Mature firms, by contrast, typically have established markets, diversified revenue streams, and organizational processes that allow for more deliberate and financially driven decision-making [147]. While both contexts must decide which features to prioritize, the weight assigned to specific criteria is likely to diverge. Although the importance of context has been widely acknowledged [148-150], there is little systematic evidence comparing prioritization criteria across startups and mature organizations. Existing studies tend to either propose frameworks [151, 152] or highlight critical success factors [31, 153], but they fall short of providing empirical insights into how practitioners in different organizational settings actually prioritize. Addressing this gap requires large-scale evidence on which criteria matter most in each context, and how those differences can inform requirements engineering and product management practices.

⁵ Authors: Frédéric Pattyn, Usman Rafiq, Xiaofing Wang, Geert Poels and Peter Goetz

To address this, we conducted a global survey of 171 practitioners from 37 countries and 51 industries, randomly assigning them to consider either a startup or a mature company context. The study is guided by the overarching research question:

RQ1. Which requirements prioritization criteria are considered most important by practitioners in startup versus mature contexts to optimize the probability of success?

This paper contributes to requirements engineering and product management research in two ways. First, it provides one of the largest empirical investigations to date of requirements prioritization criteria in practice, grounded in practitioner perspectives. Second, it informs both scholars and practitioners by clarifying which criteria should be emphasized in context-sensitive prioritization approaches, thereby advancing the alignment of decision-making methods with organizational realities.

The remainder of this paper is structured as follows. Section II reviews the related work on requirements prioritization criteria and methods, with specific attention to differences between startup and mature contexts. Section III describes the study design, including survey development, refinement, and data collection. Section IV presents the demographic profile of the respondents, while Section V reports the survey results on prioritization categories and criteria. Section VI discusses the findings in relation to prior literature, highlighting implications for startups and mature firms. Section VII outlines threats to validity, and Section VIII concludes. Finally, a Summary synthesizes the key insights, followed by a Research Agenda to guide future work.

5.2. Related Work

Despite widespread acknowledgment of the importance of Requirements Prioritization Criteria (RPC), empirical studies comparing their relevance across different organizational contexts remain scarce—particularly in the case of startups. This lack of evidence represents a critical gap in the literature. It raises fundamental questions about whether commonly used RPC are appropriate for the unique demands of startup environments. Addressing this gap requires systematic investigation into which RPC matter most in startups compared to mature firms.

The Lean Startup (LS) framework, as proposed by Ries [118], is widely recognized in product management and startup literature. It emphasizes the creation of a cost-effective Minimum Viable Product (MVP) to quickly enter the market, gather user feedback, and iteratively improve the offering. These principles encapsulate many of the contextual challenges and success factors that differentiate startups from mature companies. Building on this foundation, and informed by prior literature, the Related Work section is structured around eight dimensions relevant to requirements prioritization in startups: (A) business ideation and market validation, (B) funding, (C) cost efficiency, (D) financial performance, (E) minimum viable product (MVP), (F) time is money, (G) time to market, and (H) value. These dimensions also informed the design of the survey instrument used in this study.

5.2.1. Business Ideation and Market Validation

Building on the Lean Startup approach [124], early-stage ventures must not only test product hypotheses but also validate their broader business direction against a set of strategic and market-oriented considerations. This responsibility often falls squarely on the shoulders of the founders, who must assess whether their idea addresses a real market need [10]—a cornerstone of product-market fit [154]. The need to demonstrate the product [26] to early stakeholders further reinforces the importance of evidencing practical feasibility before substantial resources are committed. Signed contracts or letters of intent [126] can serve as critical early signals of market traction, offering founders a level of validation that can guide resource allocation and investor discussions.

Strategic clarity is equally essential. Founders must evaluate whether the envisioned solution aligns with the startup's overall strategy [79], whether it presents a strong strategic fit [10] with existing capabilities, and whether there are meaningful company synergies [10] that can be leveraged. These choices are rarely abstract; they inform what is built into the MVP, with direct implications for development scope, required capabilities, and ultimately, cost.

In parallel, market-facing factors such as competitive intensity and market maturity, including the sector's growth rate [10], must be taken into account to determine the viability and timing of market entry. Founders are also tasked with estimating the startup's potential market share [155] and assessing whether a defensible proprietary position [10] can be established. These early strategic decisions help set the stage for funding discussions and may shape investor confidence in the startup's differentiation and growth potential.

Beyond product and market considerations, founders often act as custodians of brand value [156] and cultural alignment [157], especially when entering markets where social resonance or value congruence matters. Moreover, feature promises made in early conversations with stakeholders—whether customers, investors, or partners—become commitments that inform Requirements Prioritization (RP) decisions down the line [3].

Together, these founder-led assessments comprise the foundation of business ideation and validation. They are essential for determining whether a concept warrants further investment, ensuring financial viability, and preparing for the next stage: securing funding and building the MVP with informed, cost-aware prioritization.

5.2.2. Funding

A primary challenge faced by startups pertains to securing adequate funding [26] and maintaining control over their expenditure rate, commonly referred to as the "burn rate" [56]. Notably, a significant proportion of startup failures can be attributed to cash flow constraints [56]. Consequently, it is important to recognize the critical success factors that influence a startup's potential for growth and long-term viability [153, 158]. Furthermore, contemporary investors increasingly prioritize the tangible creation of both financial and social value [159], surpassing the mere pursuit of user growth. Consequently, when startups fail to prioritize revenue generation or are unable to present a clear long-term roadmap for achieving a favorable return on investment (ROI) or internal rate of return (IRR) [35], securing subsequent funding becomes considerably more challenging [148]. For those fortunate enough to secure funding, optimizing their runway effectively revolves around two key factors: cost efficiency and financial performance.

5.2.3. Cost efficiency

Given the typically limited funding available to startups, Requirements Engineering (RE) plays a crucial role in enabling them to develop cost-efficient solutions to the practical problems faced by their users [160]. Operating under strict budget constraints [161, 162], startups are compelled to make highly strategic decisions about where to allocate resources. In this context, cost considerations [163] have emerged as dominant factors in Requirements Prioritization (RP), with expected cost [164-167] frequently cited as the most important prioritization criterion across multiple studies [163].

To maximize their operational runway, startups often prioritize RP criteria that directly reflect financial discipline, such as budget constraints [161, 168], cost benefit ratio [11, 26], efficiency [31], anticipated maintenance cost [11] and cost importance ratio [169]. These considerations stand in contrast to the prioritization strategies typically employed by more mature firms, which are often less financially constrained [128]. Consequently, a more deliberate and structured budgetary process becomes not just advisable but essential for startups [9], reinforcing the strategic integration of RE into early-stage decision-making.

This cost-sensitive prioritization mindset forms a logical bridge to the next dimension of startup viability: how such practices influence financial performance and long-term sustainability.

5.2.4. Financial performance

Giardino, Wang [28] assess the success of startups primarily in terms of their business performance, recognizing that failure to achieve financial objectives can lead to overall failure [48, 111]. Consequently, every decision-making endeavor, including those related to RP, should aim at attaining the pivotal milestone of securing the first paying customer [24], enhancing the rate of sales growth [153], optimizing profitability [31], reducing customer acquisition costs [148, 165, 170] and reaching the break-even point [24]. The ultimate aspiration is to drive genuine value creation [39] by attaining product-market fit, coupled with a sustainable revenue model [9, 31, 156], as this represents the most certain path to securing subsequent funding rounds.

Given this perspective, it is reasonable to postulate that startups, more so than established companies, are likely to consider criteria related to revenue generation and financial impact [44] when evaluating requirements. These criteria may encompass estimated expected revenue [156], profitability [125, 160, 171], break-even point analysis [172], payback period [9, 10], return on investment (ROI) [9, 173, 174], net present value (NPV) [156] and internal rate of return (IRR) [9] and cash flow analysis [155]. This shift towards a financial-centric approach may yield indirect benefits, such as improved customer satisfaction, by enabling more sustainable investment in quality, reliability, and long-term product value [175].

5.2.5. Minimum viable product (MVP)

Startups often strive to achieve a balance between cost efficiency and revenue generation by embarking on the creation of a Minimum Viable Product (MVP) [24, 125] or upcoming projects. Addressing this challenge within the realm of RPC involves the definition of the feature set coverage [26, 126, 166], focusing exclusively on the core requirements [24, 125] and their runway constraints that are most likely to elicit heightened user satisfaction [30] and value [45]. Given the advantageous synergy associated with these frugal innovations [128], it is reasonable to expect that startups would take this consideration into account when making RP decisions. Furthermore, the development of an MVP imparts momentum to a team and expedites the time-to-market [176-179] and time-to-value [45], offering additional benefits.

5.3. Time is Money

In the context of startups, where speed can determine survival, minimizing both time-to-market [34, 180]—the time required to deliver a Minimum Viable Product (MVP) or project—and time-to-value [45]—the time it takes to generate business value post-launch—becomes essential. Reducing time-to-market helps contain development costs, while accelerating time-to-value can positively influence cash flow and extend the startup's financial runway through faster user acquisition and monetization. Given these dynamics, both dimensions emerge as critical factors that should be reflected in Requirements Prioritization (RP) decisions, reinforcing the need for startups to prioritize features that accelerate learning, validation, and value delivery. This naturally leads to a deeper focus on user value and perceived usefulness in the next section.

5.3.1. Time to Market

While minimizing time-to-market is critical for startups, it is only a means to an end. The ultimate goal is to achieve time-to-value—the shortest path from product delivery to

measurable business impact. However, the ability to deliver quickly and cost-effectively depends on a number of underlying factors that shape both the feasibility and efficiency of implementation. Founders must evaluate early which requirements can be delivered within technical, resource, and regulatory constraints [181]—decisions that directly affect the startup’s capacity to generate value within a limited runway.

A first set of considerations involves technical factors and constraints. The complexity [176, 182, 183] of a feature, technical dependencies [53], likelihood of code change [182], and the re-usability or modifiability [53] of components all influence development speed and effort. In addition, choices around technology [21] stacks, compatibility requirements [182], scalability [182], and compliance with technical standards [177] can introduce friction or acceleration, depending on alignment with team expertise and architecture. These constraints require founders to assess whether technical decisions support or hinder rapid and sustainable delivery.

Secondly, risk assessment plays a pivotal role in estimating delivery feasibility, timelines and potential disruptions [53, 167, 183]. Startups must account for risks related to development bottlenecks, market uncertainty, and operational failure [164, 168, 174]. Legal and compliance risks—such as penalties [184, 185], regulations [10, 126, 186], harm avoidance [161] and legal mandates [161]—can delay progress or impose additional cost burdens. Similarly, volatility [3, 187, 188], low predictability [10], and ambiguous or frequently changing requirements increase uncertainty and reduce throughput. Conversely, prioritizing feasible [53, 165, 183], low-risk [164, 168, 174], and secure [53, 182] features can accelerate delivery without compromising product integrity.

Resource constraints and availability, especially in terms of human capital, further shape delivery potential. Founders working with lean teams must often deprioritize resource-intensive features [11, 162, 166], focusing instead on those that are faster or cheaper to build with available human resources. These choices are not merely operational—they influence which value propositions reach users, and how soon.

Lastly, quality and performance metrics directly affect the time to market, but also future tech debt. It is a continuous trade-off that the team needs to take into consideration; and how much time to invest in reducing dependencies [161, 166, 187], similarities [183] and ambiguity of features [53, 183], elaborate documentation [182], and efforts to reduce errors and future rework [31, 179]. In fast-paced environments, building with testability [53], traceability [53], durability [10], and clear non-functional characteristics [21, 189] are also part of this constant deliberation between speed and accrued tech debt after release.

Together, these technical and organizational factors define the boundaries of what can be delivered, when, and at what cost. However, fast delivery alone is insufficient. What ultimately matters is delivering features that users perceive as valuable. The next section explores how startups can prioritize requirements that maximize user value—and, in doing so, accelerate their path to product-market fit and financial viability.

5.3.2. Value

User value [24, 125] is intrinsically linked to the prioritization of software project objectives. This makes it the most commonly and generically referenced requirements prioritization criterion for determining the next development steps [31, 153] because it covers so many different definitions. Ideally value is considered an outcome that has a direct impact on the valuation of the venture, or its financial runway, essentially achieving product market fit, although, in literature it also covers a lot more generic criteria such as stakeholder preferences [21], customer value [125], importance [161, 169, 185, 187, 190], user friendliness [182], user engagement [31], frequency of use [161], contribution to user task [178], and user loyalty [156]. Understanding of the objective of generating user value can be determined by objective

data point measurements, but also through the validation of customer value occurs through the collection of user feedback following product release [30].

Beyond startup-specific prioritization criteria, prior research has also investigated requirements and service ranking under uncertainty using hybrid and probabilistic approaches, particularly in cloud computing contexts, demonstrating how structured ranking methods can support decision-making when user requirements are incomplete or uncertain, albeit in domains and organizational settings that differ from software startups and mature product organizations [191, 192].

5.4. Research Methodology

5.4.1. Survey design

The survey instrument was developed to capture practitioner perspectives on requirements prioritization in both startup and mature company contexts. It consisted of two main questions.

Question 1. Respondents were asked: “Select the two most important requirements considering your context to optimize the probability of success of the venture.” They were also asked to identify the criterion they considered least important. The available responses were drawn from ten requirements prioritization categories defined by Svahnberg, Gorschek [21].

Question 2. To complement and validate Question 1, participants were asked an open-ended question: “What does the ideal process look like to prioritize all existing ideas to determine what will actually be considered for the roadmap (consider timing, selection variables, method ...), and why?” This question served as a consistency check on the structured responses while also generating qualitative insights into practitioners’ reasoning and preferred approaches.

To ensure transparency and reproducibility, the survey also included a comprehensive demographic section. Basic demographic variables included gender, age, highest educational degree, country of residence, and primary language. In addition, practitioner-specific variables captured years of experience in product roles, years as a product manager, and whether respondents were currently working on a software or SaaS product. To contextualize startup-specific experience, the survey asked whether participants had worked in startups, how many they had been involved with, the duration of their involvement, and whether they had served as part of a founding team. These measures were included to strengthen the interpretability of results and to allow analyses across relevant subgroups.

5.4.2. Survey validation and pilot testing

Before involving human experts, an initial exploratory pre-test (PT) was conducted using three AI-generated personas (PT1–PT3), created with ChatGPT and DALL·E. These personas were constructed to reflect the demographic and professional characteristics of the intended respondent population (e.g., product managers, founders, software engineers). Their simulated responses (Table 23) were used exclusively to refine question wording, reduce complexity, and estimate completion time. No AI-generated responses were included in the empirical dataset.

The revised draft survey was then reviewed by an academic expert with extensive experience in survey methodology and requirements engineering. This expert assessment led to further improvements in clarity, coverage, and response option formulation.

Subsequently, the survey underwent a practitioner pilot test during the PR_D_CT DAY conference hosted by Sirris in Ghent, Belgium (September 11, 2023), where approximately 120 product managers were invited to participate. Three pilot responses (P1–P3; Table 24) were collected and used solely to validate interpretability, usability, and question sequencing; these responses were not included in the final dataset.

Through this iterative process—AI-assisted exploratory pre-testing, expert review, and practitioner pilot testing—the survey instrument was progressively refined to enhance clarity, ensure construct validity, and minimize respondent burden.

Table 23: Pre-test (PT) survey answers

ID	Q1	Q2
PT1	Technical Factors and Constraints	I would look at the data we have regarding our business and customers first. We will have this maybe from quant/qual research, maybe from trackers, sales records, and so on. Judging by this we'd be able to see what the valuable items are. And then you consider about the complexity. You talk with your developers/engineers to get their opinion on the matter. And then consider the urgency as well, for example if there is a new regulation that forces the company to make some changes to the software, then that should be implemented ASAP. This is in my opinion a good way to prioritize because it is data-driven, considers complexity/feasibility, as well as urgency/timing.
PT2	Stakeholder and Customer Considerations	Determine which initiatives cover multiple product areas: business, vision, customer, strategy. Then prioritize based on rice.
PT3	Revenue and Financial Impact	In all cases, it depends on the severity of the problem that the idea is solving and the potential for ROI. Going deeper into the different ideas depends on whether it is a new idea (v1) or an improvement on the existing product (v2+). For new ideas, it is essential to build something small, ship it, and evaluate it. Why? Because empirical evidence is the only that counts. The skill is to learn to make something small enough. Prioritization of new ideas should not be overcomplicated. Timebox it and go to work.

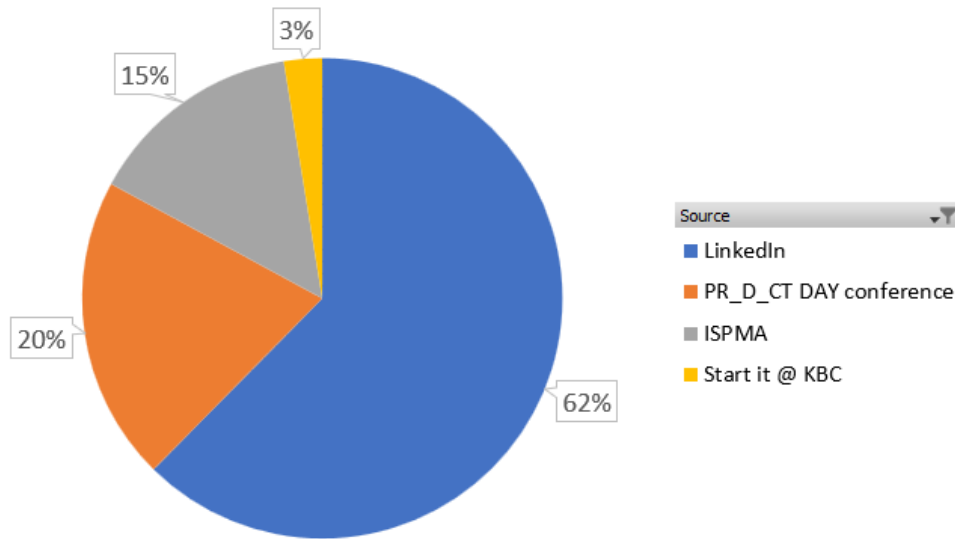
Table 24: Pilot (P) survey answers

ID	Q1	Q2
P1	Business and Strategic Impact	What is our vision => what is the strategy to achieve that vision => what are the objectives to achieve that strategy? => analyze data, talk to users and stakeholders
P2	Cost-Related Factors	Empty
P3	Business and Strategic Impact	allocating a % of resource to 2-5 pre-selected areas of the business (about 1x per year based on the overall strategy) and then optimize within each area. The latter can be done by the PM and development team in order to reach an objective.

5.4.3. Participants

In total, 171 valid responses were collected between 10 November 2023 and 2 January 2024 using a convenience sampling method [193]. A response was considered valid when the participant completed the core survey questions related to requirements prioritization. The survey was distributed through several professional networks, including the International Software Product Management Association (ISPMA®), Blackblot®, the Product Consortium, and Start it @ KBC. The majority of respondents were reached through LinkedIn outreach, through which 664 individuals were contacted. Of these, 412 (62%) viewed the invitation message and 182 provided either positive or negative feedback (Figure 12). This resulted in 107 complete responses—i.e., those who also provided full demographic data—corresponding to a feedback rate of 27.4% and a completion success rate of 16.1% for the LinkedIn channel.

Figure 10: Sources of survey respondents



5.4.4. Data Collection & GDPR compliance

The survey was voluntary and uncompensated. At the beginning of the questionnaire, respondents were provided with an information sheet in accordance with the General Data Protection Regulation (GDPR). This document outlined the purpose of the research, the contact details of the responsible researchers, and an explicit consent request. Participants were informed that their responses would be anonymized, used solely for academic purposes, and that they could withdraw at any time. Only after providing consent were they randomly assigned to consider either a startup or mature company context when answering the questions.

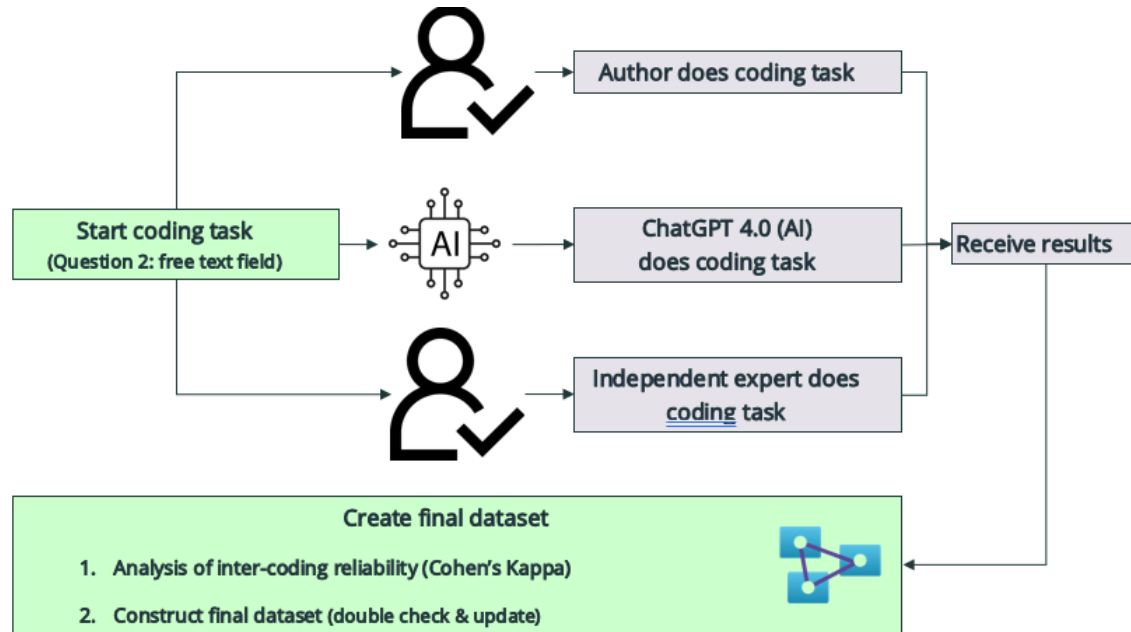
5.4.5. Data Analysis

Question 1 generated structured, categorical data that were analyzed using descriptive statistics and comparative analyses across the startup and mature contexts. Question 2, being open-ended, required qualitative analysis. A hybrid coding approach was applied (Figure 13), combining inductive and deductive techniques as outlined in prior work [141].

To enhance robustness, the initial coding was performed independently by the first author and an AI-assisted coder (ChatGPT 4.0), after which an independent human expert conducted a third coding pass. The use of an AI-assisted coder served as an exploratory support mechanism consistent with emerging hybrid coding practices [141], while all final coding decisions were reviewed and validated by human coders.

Inter-coder reliability was assessed on a subset of responses, yielding Cohen’s κ scores of 0.49 and 0.48, indicating a fair level of agreement [194]. Given the exploratory nature of the study and the multi-coder validation process, this level of reliability was deemed acceptable.

Figure 11: Hybrid coding process



5.4.6. Limitations

As with all survey-based research, certain limitations apply. The reliance on voluntary participation may introduce self-selection bias, and the use of professional networks may overrepresent particular industries or regions. In addition, while random assignment to startup versus mature company contexts supports comparative validity, responses are based on perceived rather than observed decision-making behavior.

5.5. Results

Demographic data from survey

The final dataset consisted of 171 valid responses, representing 37 countries and 51 industries. Respondents included product managers, software engineers, founders, and entrepreneurs. On average, participants reported six years of product management experience, and 36.8% had founded at least one startup. Educational backgrounds ranged from bachelor’s to doctoral degrees, and respondents were evenly spread across age groups, ensuring a diverse sample suitable for comparative analysis. A summary of respondent demographics is shown in Table 43.

Table 25: Demographic Profile of Respondents (n=171)

Characteristic	Distribution
Gender	29.8% female, 70.2% male
Average age	38 years overall (35 years for women; 40 years for men)
Founding experience	36.8% have founded ≥ 1 startup

Product management experience	Mean = 6 years
Countries represented	37 countries
Top countries	Belgium (40.4%), United States (8.8%), India (6.4%), Germany & UK (4.7% combined)
Industries represented	51 industries
Top industries	FinTech (16.8%), HR Tech (11.0%), IT services (9.5%)

5.5.1. Research Question 1: “Which requirements prioritization categories and criteria are considered more important by practitioners in a startup context, compared to mature context to achieve success?”

Requirements Prioritization Categories

To understand how practitioners weigh Requirements Prioritization Categories (RPCs) in startup versus mature environments, we analyzed both the structured Q1 responses and the open-ended Q2 responses. Q1 captures explicit, surface-level prioritization under fixed options, while Q2 captures implicit, process-driven prioritization expressed in respondents’ own terms. Differences between Q1 and Q2 are therefore expected and informative. Figures 14 and 15 present the raw counts and proportional selections of each RPC category across startup and mature contexts for Q1 and Q2.

Figure 12: Counts and proportional selections for Q1 (“Most Important” RPC Category)

Category	Startup	Mature	Proportion for Startup	Proportion for Mature	p-value
Business and Strategic Impact	20	8	0.2985	0.2857	
2 Cost-Related Factors	7	1	0.1045	0.0357	
3 Quality and Performance Metrics		1		0.0357	0.0000
4 Resource Constraints and Availability	1		0.0149		
5 Revenue and Financial Impact	6	4	0.0896	0.1429	0.0000
6 Risk Assessment	2		0.0299		0.0000
7 Stakeholder and Customer considerations	24	12	0.3582	0.4286	0.6926
8 Technical Factors and Constraints	2		0.0299		0.0000
9 Time Constraints and Scheduling	5	2	0.0746	0.0714	0.3837

Figure 13: Counts and Proportional Selections for Coded Q2 (“Ideal Process”) Categories

Category	Startup	Mature	Proportion for Startup	Proportion for Mature	pvalue
1 Business and Strategic Impact	30	15	0.2609	0.3061	
2 Cost-Related Factors	3	3	0.0261	0.0612	0.0000
3 Dependency and Interdependency Factors		1		0.0204	
4 Quality and Performance Metrics	2	4	0.0174	0.0816	0.0000
5 Resource Constraints and Availability	1	1	0.0087	0.0204	
6 Revenue and Financial Impact	26	10	0.2261	0.2041	0.8136
7 Risk Assessment	4		0.0348		0.0000
8 Stakeholder and Customer Considerations	41	13	0.3565	0.2653	0.8679
9 Technical Factors and Constraints	5	1	0.0435	0.0204	0.4597
10 Time Constraints and Scheduling	3	1	0.0261	0.0204	0.0000

To quantify the relative emphasis between contexts, we computed a context score for each category.

$$\text{Context Score} = \frac{\text{Proportion of selections in the startup context}}{\text{Proportion of selections in the mature context}}$$

Values > 1 indicate greater emphasis in startups, while values < 1 indicate greater emphasis in mature firms. Categories selected exclusively by one group are reported as startup-only or mature-only. Example based on the Cost-Related Factors category with a context score of 2.93 from Figure 15.

- Startup proportion = 0.1045
- Mature proportion = 0.0357
- $0.1045/0.0357 = 2.93$

The resulting context scores are shown in Table 44.

Table 26: Context scores for RPC categories in Q1 and Q2

RPC category	Q1	Q2
Business and Strategic Impact	0.85	1.04
Cost-Related Factors	0.43	2.93
Dependency and Interdependency Factors	(1)*	
Quality and Performance Metrics	0.21	(1)***
Resource Constraints and Availability	0.43	(1)****
Revenue and Financial Impact	1.11	0.63
Risk Assessment	(4)**	(2)*****
Stakeholder and Customer Considerations	1.34	0.84
Technical Factors and Constraints		(2)*****
Time Constraints and Scheduling	1.28	1.04

*a single observation in favor of a mature context

** four observations, all in favor of a startup context

*** a single observations in favor of a mature context

**** a single observation in favor of a startup context

***** two observations in favor of a startup context; 0.0299 vs 0.00

***** two observations in favor of a startup context; 0.0299 vs 0.00

To determine whether the directional differences observed in Q1 (the structured “most important” selection) are statistically significant, logistic regression models were estimated for each RPC category. In these models, the dependent variable was the assigned context (startup = 1, mature = 0), and the independent variable was the selection of each category in Q1. The resulting p-values are shown in Table 45.

Because Q2 is an open-ended qualitative question that was subsequently coded into categories, it does not yield binary selection variables suitable for logistic regression. As a result, statistical significance is assessed only for Q1.

Table 27: Logistic regression p-values and context score for RPC category selections (Q1)

RPC Category	Q1 p-value	Context Score
Cost-Related Factors	0.00	2.93
Quality and Performance Metrics	0.00	0.21
Revenue and Financial Impact	0.81	0.63
Risk Assessment	0.00	Startup context only
Stakeholder and Customer Considerations	0.87	0.84
Technical Factors and Constraints	0.46	Startup context only
Time Constraints and Scheduling	0.00	1.04

Although, when drilling down, by consulting Table 44 and Table 45, it appears that this category is relevant in both contexts. The score of 0.43 for Q1 means almost twice as important for a mature context, while in contrast, according to the Q2 value, with 2.93, it is almost three times as important in a startup context. When considering this category has also a 0.00 p-value as the least important category, combined with its contextual value of 0.76, this may indicate that its importance leans slightly toward a startup context.

Quality and Performance Metrics are a second RPC category that seems to be very important based on the p-values (Table 45) of both questions. When looking at the context specific results, it can be observed that it is really a category that's mainly important for the mature context, considering a 0.21 score for Q1 and only having observations for the mature context for Q2 (Table 44).

Revenue and Financial Impact responses are inconsistent on the level of importance. When having to select it from a dropdown (Question 1), it doesn't seem to be as important as when they can share their ideal process freely (Question 2). Still, when considering Q2 (the coded 'ideal process' responses), and the subgroup analysis of prior founders ($p = 0.19$ in Q1), it is interesting to examine which context this category appears more salient (Figure 17). With a 1.11 value for Q1 and 0.63 for Q2, and only having observations for this category in a startup context as the least important category, it could give more credence to being more important within a mature context.

The Risk Assessment category also appears to be highly salient, with a p-value of 0 in Q1 (Table 45), and consistent prominence in the coded Q2 responses. For both Q1 and Q2 (Figure 17) there are only observations noted within a startup context. Still, this initially obvious importance as a criterion within the startup context needs to be nuanced as it is 1.8 times as much mentioned in a startup context as the least important category in comparison with in a mature context.

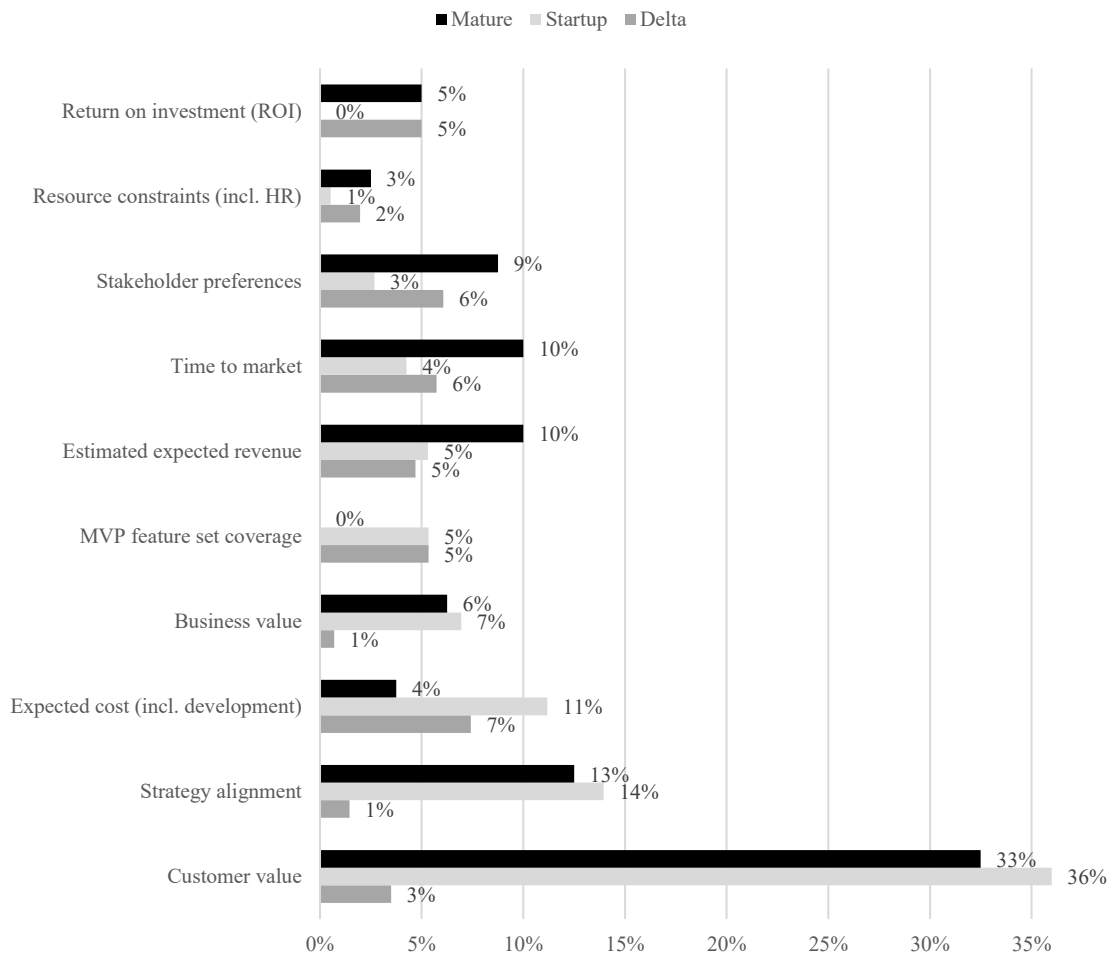
Technical Factors and Constraints can also be considered mixed with a p-value of 0.46 for Q1, while appearing prominently in the coded Q2 responses based on a limited number of startup-context observations (Figure 17). Taking further into account these results, its value within the startup context gets nuanced even more, considering that the category has been selected more in the startup context with a p-value of 0.00 as one the least important category to take into consideration for prioritization.

Time Constraints and Scheduling has not got consistent results between both questions, but with a p-value of 0 for Question 1, and a noteworthy 0.25 among those respondents that already have been a founder in the past, it is interesting to check for which context it might be more important (Figure 17). There is a slight preference towards the startup context with a value of 1.28.

Requirements Prioritization Criteria

The descriptive analysis of the answer to the requirements prioritization criteria (RPC) data (Figure 16) indicates that Customer Value has been mentioned the most as the most important requirements prioritization criterium across both contexts. With such a high percentage, it might be considered a general staple in the used RPC as part of the decision-making process, but without holding any significance for any context.

Figure 14: Requirements Prioritization Criteria: Mature companies vs Startups



“Expected cost” as requirements prioritization criteria (RPC) seems by far the most significant criteria for startups. This by combining a p-value (1) of 0.05, with a 2.15 context score (Figure 17). The latter signaling that it is more than twice as important for early stage ventures than for those that are more mature. Only two other criteria seem to be of interest: MVP feature set coverage and Time to Market. The former has a p-value (1) of 0.36 and gets heavily supported by those respondents that have been founders in the past and, or currently are product managers. The latter has a p-value (2) of 0.54 which indicates it is almost equally important for both contexts, but does get nice support with a 0.12 p-value by those respondents that are currently product managers.

Figure 15: P-values for RPCs and Contextual Factors

RPC	p-value (1)	p-value (2)	Startup Context	Times Founder	Currently Product Manager
Complexity	1.00	0.00	(1)*	0.99	0.00
Cost importance ratio	1.00	0.00	(2)**	0.00	0.00
Customer value	0.84	0.00	1.14	0.19	0.79
Dependencies	1.00	0.00	(1)***	0.00	0.00
Estimated expected revenue	0.81	0.00	0.29	0.88	0.18
Expected cost	0.05	0.00	2.15	0.41	0.75
MVP feature set coverage	0.36	0.00	(4)****	0.00	0.00
Return on investment (ROI)	1.00	0.00	(2)*****	0.00	0.00
Risks	1.00	0.00	(2)*****	0.00	0.00
Stakeholder preferences	0.98	0.00	0.29	0.94	0.67
Strategy alignment	0.78	0.00	1.18	0.41	0.29
Time to market	0.54	0.00	0.43	0.70	0.12

(1): p-value calculated by combining the observations of the “main” and “second” criteria. The lower the p-value, the more significant for the startup context.

(2): p-value calculated by taking into account only the observations of the “main criteria.”

* a single observation in favor of a startup context; 0.0154 vs 0.00

** two observations in favor of a startup context; 0.0308 vs 0.00

*** a single observation in favor of a startup context; 0.0154 vs 0.00

**** four observations, all in favor of a startup context; 0.0615 vs 0.00

***** two observations in favor of a mature context; 0.00 vs 0.0714

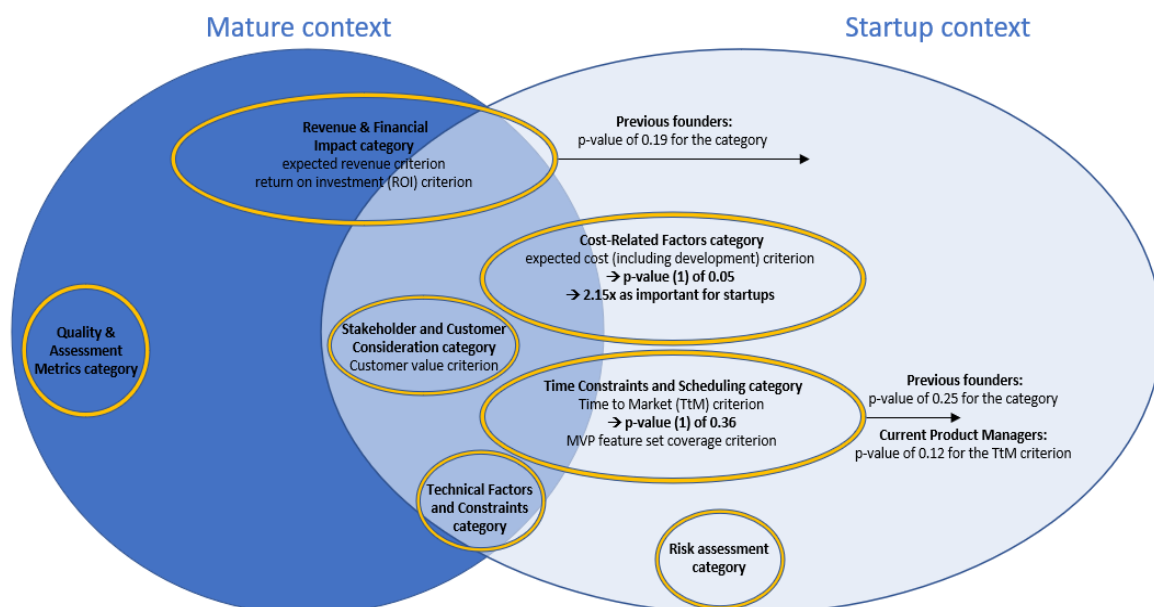
***** two observations in favor of a startup context; 0.0308 vs 0.00

Cross-Validation between categories

As shown in Figure 18, there is substantial overlap between the findings of both research questions. Several requirements prioritization criteria (RPCs) emerge consistently across Q1 and Q2, with some becoming even more pronounced when respondents describe their prioritization logic in their own words.

The results show that startups place notable emphasis on cost-related factors—including expected revenue and return on investment (ROI)—whereas mature companies consistently foreground quality and performance metrics. This is further supported by the observation that previous startup founders also view cost-related criteria as important ($p = 0.19$). Taken together, these analyses highlight clear differences in strategic considerations and decision-making processes between startups and mature firms, offering actionable insights for refining requirements prioritization approaches in different contexts.

Figure 16: Overlapping of criteria between startups and mature companies



5.6. Discussion

5.6.1. Cost efficiency

The study underscores the paramount importance of cost efficiency for startups, a finding consistent with the high failure rates attributed to cash flow challenges. By placing a significant emphasis on cost-related factors such as expected development cost, startups can extend their operational runway and reduce the risk of insolvency. This strategic focus on cost efficiency is not merely about minimizing expenses but optimizing the allocation of limited resources to ensure sustainability. Our data, showing that cost efficiency is a critical differentiator between successful and unsuccessful startups, reinforces the necessity for startups to rigorously manage their burn rates and prioritize low-cost, high-confidence features.

5.7. Revenue and financial impact

The study offers intriguing insights into the financial decision-making approaches of both types of companies. While the literature suggests that startups should prioritize decisions that enhance revenue generation, like acquiring the first paying customer, improving sales growth, maximizing profits, reducing acquisition costs, and achieving a sustainable revenue model, because such strategies are deemed essential for securing future funding rounds and ensuring business viability, study results are more nuanced.

The survey results indicate inconsistency among the respondents. Only when they freely think about their ideal requirements prioritization process, the Revenue and Financial Impact RPC category gets considered more important for startups, especially by those respondents that have prior entrepreneurial experience; it is notable that Return on Investment (ROI) or expected revenue criteria are exclusively mentioned by respondents in the mature company context. This observation would mean that established companies have more to gain by prioritizing their features based on financial criteria in their decision-making processes than startups.

When considering the results of the case studies, it can be observed that those startups that do consider the business case in their prioritization efforts, such as including expected revenue or the ROI of the feature, correlate with future positive results. Considering that the tier 1 and 2 startups both do it approximately five more often than the less unfortunate ones, the highest discrepancy among all criteria.

While startups are theorized to be more financially driven in their decision-making, our findings indicate that overall established companies might actually place a higher emphasis on direct financial outcomes, and only a very small group of startups. This inconsistency among the early stage ventures presents a fascinating area for further research. Although financial goals are acknowledged as critical in the literature, the nuanced ways startups prioritize them in decision-making are not yet fully understood, although this study shows that understandably it might be highly beneficial to do so. Understanding these differences in greater detail could uncover the underlying factors that drive startups' decision-making strategies, providing insights that can help refine financial strategies tailored to startups' unique challenges and opportunities. A single hypothesis could be that, due to the inherent uncertainty of the startup context forces founders to focus on the criterion they can control most: cost, in comparison to revenue. Further research into this phenomenon could illuminate how startups can better balance financial priorities to enhance their chances of success.

Revenue generation and financial impact are pivotal in determining the long-term success and viability of startups. Our findings indicate that successful startups prioritize features that have a solid business case, including criteria such as expected revenue, ROI, and profitability. This financial-centric approach helps startups secure the crucial initial funding rounds and build a sustainable business model. Notably, the study reveals that those startups are more likely to achieve financial milestones and attract further investment. This approach not only enhances

their financial performance but also builds credibility with investors, thereby facilitating continued growth and development.

5.7.1. Minimal Viable Product (MVP)

The concept of the Minimum Viable Product (MVP) reveals interesting nuances in the requirements prioritization processes of startups versus established companies. Literature suggests that startups often strive to combine cost efficiency with revenue generation by focusing on MVP development. The critical RPC in this context involves defining an MVP feature set that encompasses core requirements likely to maximize user satisfaction.

Our research consistently indicates that mature companies place less emphasis on MVP feature set coverage, as reflected in our data. In contrast, startups prioritize MVPs due to their need for agile, cost-effective development, and the support of their founders, as reflected in our criteria-level results (Figure 13). This focus on MVPs in startups is influenced by factors like resource constraints, technical limitations, and dependencies, as also reflected in the cross-validation results (Figure 14). The difference suggests that mature companies, with their established market presence and resources, often bypass the MVP stage, while startups rely on MVPs to quickly validate product-market fit. The declining emphasis on MVPs with experience reflects a shift toward more extensive product development strategies as startups mature.

This insight highlights the importance of further research into what makes an MVP effective and the decision-making processes behind it. Based on the interviews it can be observed that the tier 1 startups put a lot of time and effort in the validation of their ideas before building the actual MVP. This is a great way to increase their confidence level about their ideas and potential product market fit. All of them would do even more research if they could start over, and none of them would start building without having secured paying customers. Increasing their confidence level by these small effort iterations to refine the product before moving forward with building a full MVP could be introduced as Minimal Viable Experiments (MVE). This way of working provides rapid feedback loops, essential to increase further the confidence levels, and decrease further the risk of the early stage venture. By investigating how startups define and prioritize MVP features, researchers can identify key factors that drive successful product-market fit and agile development. This can help refine the understanding of MVP strategy, ultimately guiding startups to make more informed decisions that align with their unique constraints and ambitions in the early venture stages.

5.7.2. User value

Delivering User value is a cornerstone of successful startup strategy. The study indicates that startups which prioritize customer value, user satisfaction, and user engagement are more likely to succeed, but only when this is based on quantitative measures (count of requests) and not just informal discussions and opinions. By incorporating customer feedback into the development process and aligning product features with customer needs, startups can build loyal user bases and drive long-term engagement. The data highlights that startups need to balance stakeholder preferences with objective user data, ensuring that their product development efforts are both market-driven and user-centric. This alignment with customer needs not only enhances user satisfaction but also drives revenue growth and reduces the risk of product failure.

5.7.3. Practical Implications

For startup product managers and founders, the findings highlight the importance of explicitly prioritizing cost efficiency and confidence-building mechanisms in early-stage decision-making. Rather than relying primarily on projected financial returns, startups should focus on minimizing expected development cost, and validating assumptions through small, iterative

experiments before committing significant resources. Practices such as defining lean MVP scopes, using minimal viable experiments, and grounding prioritization decisions in measurable customer feedback can help extend runway while reducing uncertainty.

For mature organizations, the results suggest that requirements prioritization should more explicitly leverage structured financial criteria such as expected revenue, ROI, and profitability. Unlike startups, mature firms benefit from established markets and data availability, allowing them to systematically evaluate financial impact and quality metrics when making prioritization decisions. Emphasizing these criteria can support portfolio-level optimization and ensure alignment between product roadmaps and long-term business objectives.

For the design and selection of prioritization methods, the clear differences observed between startup and mature contexts indicate that “one-size-fits-all” approaches are suboptimal. Practitioners should select or tailor prioritization techniques based on organizational context: lightweight, cost- and learning-oriented methods for startups, and more financially and quality-driven methods for mature companies. Tool vendors and method designers should incorporate context sensitivity as a core design principle rather than treating it as an optional extension.

5.8. Threats to Validity

Several potential threats to validity must be considered when interpreting the results of this study.

Sampling bias and external validity. Responses were collected through professional networks and LinkedIn outreach, which may have led to an overrepresentation of certain geographies and industries. For example, Belgium accounted for over 40% of respondents, with FinTech and HR Tech among the largest industry groups. While the sample included 171 practitioners across 37 countries and 51 industries, these imbalances may limit the generalizability of the findings to the broader startup and product management population.

Non-response bias. Because the majority of responses were gathered through LinkedIn, participants who are active on professional platforms may differ systematically from those who are not. Non-respondents may also hold different prioritization practices, potentially skewing results.

Construct validity. The survey questions were designed to capture the perceived importance of requirements prioritization categories (Q1) and the reasoning behind prioritization processes (Q2). However, self-reported perceptions do not necessarily reflect actual prioritization behavior in practice. Moreover, differences in the structured (dropdown) and open-ended formats may have introduced inconsistencies in how respondents interpreted and applied the criteria.

Self-reporting bias. As with most survey research, responses reflect what practitioners say they do or believe is important, rather than what they demonstrably do. Social desirability bias may have influenced some responses, particularly in emphasizing customer value or strategic alignment.

Coding subjectivity. The open-ended responses required qualitative coding. Although two independent coders applied a hybrid inductive–deductive process, inter-coder reliability scores ($\kappa = 0.49$ and 0.48) indicate only a fair level of agreement. While this level is acceptable for exploratory research and consistent with other software engineering survey studies, the interpretation of qualitative themes should be regarded with appropriate caution.

Mitigation measures. To address these threats, the survey employed random assignment to startup versus mature contexts, combined structured and open-ended questions to cross-validate responses, and used multiple coders to enhance reliability. GDPR-compliant consent procedures and voluntary participation reduced ethical risks but may have amplified self-selection effects.

Overall, while these limitations constrain the generalizability and precision of some findings, the study offers valuable exploratory insights into how requirements prioritization differs between startups and mature organizations.

5.9. Conclusions

This study set out to examine which requirements prioritization criteria matter most in startups compared to mature companies. The survey findings demonstrate that organizational context profoundly shapes prioritization choices. Startups emphasize cost efficiency, time-to-market, and MVP feature coverage, reflecting the acute pressures of uncertainty and survival. Mature organizations, by contrast, lean more heavily on financial returns and quality assurance, consistent with their stability and optimization focus.

These results challenge the assumption that startups are primarily financially driven. Instead, they highlight a dominant cost-control mindset, with limited attention to formal financial criteria such as ROI or NPV. This underlines a paradox: while startups face severe cash flow risks, they often neglect systematic financial analysis in their decision-making. Conversely, mature companies may overemphasize financial metrics, potentially at the expense of agility. The contribution of this study lies in providing empirical evidence for these contextual differences, offering one of the largest comparative datasets on requirements prioritization to date. For researchers, the findings suggest that prioritization methods must be tailored to context rather than treated as universal. For practitioners, they underscore the need for startups to complement their cost-driven mindset with structured business-case thinking, and for mature firms to safeguard agility while pursuing financial optimization.

5.10. Summary

This survey of 171 practitioners across 37 countries and 51 industries provides one of the most comprehensive empirical comparisons of requirements prioritization practices between startups and mature organizations. The findings reveal clear contextual contrasts. Startups emphasize expected cost, assigning it more than twice the importance compared to mature firms (context score 2.15). They also highlight the importance of time-to-market and MVP feature coverage, reflecting the urgency of survival, rapid validation, and resource constraints. Mature organizations, by contrast, prioritize revenue generation, return on investment (ROI), and quality metrics, aligning with their focus on financial optimization and established processes.

Across both contexts, customer value consistently ranks among the most important criteria (cited by over 30% of respondents). However, its operationalization differs: startups often rely on informal validation through quick iterations and user feedback, whereas mature companies incorporate structured assessments supported by financial and quality data.

Taken together, these results provide actionable insights for practitioners and a robust evidence base for researchers. They underscore the need for context-sensitive approaches to requirements prioritization: startups benefit from disciplined cost and speed considerations, while mature companies gain from structured financial and quality-driven prioritization. By clarifying these differences, the study offers both academics and practitioners a more nuanced understanding of how prioritization decisions are shaped by organizational context.

5.11. Research Agenda

Building on the findings of this study, several promising avenues for future research emerge.

1. Defining the MVP requirements set.

Our results confirm that MVP coverage is a critical prioritization criterion for startups, yet the processes behind defining MVP scope remain underexplored. Future research should examine best practices for determining MVP feature sets, including decision variables, stakeholder

involvement, and validation strategies. Such insights would provide startups with practical guidance to tailor MVPs for rapid market entry while safeguarding cost efficiency, for example through longitudinal case studies of early-stage startups comparing MVP scope decisions with subsequent validation outcomes such as user adoption, iteration speed, and cost overruns.

2. Financial decision-making under uncertainty.

Contrary to expectations in the literature, startups in our study did not consistently prioritize financial criteria such as ROI or NPV, whereas mature companies did. This paradox suggests that startups may emphasize cost because it is more controllable under uncertainty. Future work should investigate how startups balance cost, revenue, and financial forecasting, and whether more structured financial prioritization could improve survival rates. Such trade-offs could be examined through controlled experiments or retrospective analyses comparing startups that apply structured financial prioritization techniques with those relying on informal cost-based heuristics.

3. Toward context-specific prioritization methods.

The contrasts identified between startups and mature firms underscore the need for requirements prioritization methods that are explicitly tailored to organizational context. Future research should explore how such methods can integrate context-specific criteria—such as cost efficiency, time-to-market, and MVP validation for startups—while maintaining flexibility for mature settings. Future studies could operationalize this by designing and evaluating prototype prioritization methods in real-world startup and mature company settings, assessing their impact on decision speed, alignment, and project outcomes.

4. Emerging technologies as enablers.

Technologies such as artificial intelligence and blockchain are rapidly transforming startup practices, including product management and requirements engineering. Investigating how these technologies can support or automate prioritization decisions represents another promising direction. This could be investigated by evaluating AI-assisted or rule-based prioritization tools in experimental or field settings, comparing human-only versus technology-supported decision-making processes.

5. Improving generalizability.

Finally, future studies should expand beyond convenience sampling to include more diverse respondent populations across geographies and industries. Combining quantitative survey data with qualitative insights from case studies would strengthen validity and provide richer understanding of prioritization practices. In particular, mixed-method research designs combining large-scale surveys with in-depth case studies across multiple regions would enable stronger triangulation and external validity.

By pursuing these research directions, the field can move toward a more nuanced and context-sensitive understanding of requirements prioritization, ultimately equipping startups and mature organizations with decision-making approaches better aligned with their realities.

6. Financial and Temporal Decision Criteria in Software Startup Prioritization: Introducing Time to Value (TtV) and Reassessing ROI through Founders' Insights vs. Systematic Literature Review

Abstract⁶

Software startups operate under extreme uncertainty, short time horizons, and limited financial resources—conditions that demand rapid, evidence-based product decisions. Yet, most existing Requirements Prioritization Criteria (RPC) frameworks remain rooted in stable enterprise environments, emphasizing process consistency over survival dynamics. This study combines a systematic literature review of 40 studies with 34 founder interviews across 19 software startups to examine how prioritization criteria are defined, applied, and related to performance. The review uncovered 82 unique criteria, revealing strong fragmentation and limited consideration of startup-specific contexts. Empirical results show a substantial overlap with the literature but also expose clear gaps: founders systematically apply financial and temporal criteria—notably expected cost, business case strength, and the newly identified Time to Value (TtV)—while these remain virtually absent from academic models. Startups that explicitly used these criteria achieved stronger financial and operational performance, whereas those neglecting them often failed. This finding underscores that financial discipline and time-sensitive decision-making are predictive of startup survival. The study contributes by bridging academic and entrepreneurial perspectives on prioritization and by proposing TtV as a critical addition to RPC frameworks—highlighting the need for methods that integrate speed, cash efficiency, and value realization into the core of requirements engineering for software startups.

Keywords

Startups, Requirements Prioritization Criteria, Requirements Engineering, Product Management, Time to Value (TtV), Financial Criteria, ROI, Decision-Making

6.1. Introduction

Software startups are notorious for operating in a high-paced, uncertain, and resource-constrained context [24], faced with numerous challenges contributing to their high failure rates, including premature scaling, cash flow challenges (82%), difficulties in obtaining financing or investors (47%) [50, 98] and funding shortages (21 to 44%) [51, 100]. A significant proportion, up to 63% fail [51], with a quarter of these failures occurring in the first year alone [100]. Therefore their survival is tightly linked to prioritizing those features that will mitigate of much of these challenges as possible.

In their early stages, startups often embark on product decision-making activities [49] without a strategic plan for product development [48]. This lack of foresight can lead to inefficient requirements selection process [29], misallocation of critical resources [57], insufficient market research or business case analysis [50], and an over-commitment to solutions that do not meet market needs [122]. One example is the tendency to invest heavily in a feature-rich Minimum Viable Product (MVP) with a long time-to-market [195], often without traction or a clear business development strategy [24]. Consequently, they face the risk of not achieving a product-market fit [49] and struggle with acquiring their first paying customers [24], negatively affecting the startup's runway (number of months a company has left until cash runs out) and therefore probability of success.

Effective cash flow management, driven by strategic planning and efficient resource allocation, emerges as a critical factor in ensuring future venture survival. However, there exists a significant gap in academic literature regarding the analysis of cash flow as a determinant in requirements prioritization, particularly within the startup context. Existing research, while highlighting the significance of financial ratios in requirements prioritization discussions, often overlooks the unique challenges faced by startups. Furthermore, the distinct conditions and constraints under

⁶ Authors: Frédéric Pattyn and Geert Poels

which startups operate necessitate a tailored approach to requirements prioritization. The absence of specific research underscores the importance and urgency of conducting this study to address identified gaps [44].

In the end it all boils down to managing as good as possible the startup's runway [56, 196], the number of months a company has left until their cash runs out; trying to keep their costs as low as possible, as long as needed until they convince their stakeholders to give money, whether as paying customers, or investors [197, 198]. The survival of startups is highly dependent on their ability to strive for these results [199].

This study aims to combine a structured literature review and a multiple-case study to construct a unified analysis of requirements-prioritization criteria in software startups. The goal is twofold: first, to contrast insights from the literature with evidence from practice to identify potential new or context-specific criteria that are relevant in startup environments; and second, to examine whether the use of financially oriented prioritization criteria is associated with stronger startup performance.

This work contributes by: (i) presenting a consolidated catalogue of requirements-prioritization criteria tailored to the startup context (grounded in prior literature and contrasted with practice); (ii) documenting criteria that emerge from real-world cases and are under-represented in the literature; and (iii) providing initial, association-level evidence on how financially oriented criteria relate to indicators of startup performance—offering practical heuristics for founders and a clearer agenda for subsequent, confirmatory research.

To achieve these aims, we conduct a structured literature review and a multiple-case study of software startups, then synthesize findings through cross-case and cross-source comparison to align (or contrast) criteria identified in theory with those observed in practice.

The remainder of this paper is organized as follows: Section 2 reviews related work on requirements prioritization methods and positions the present study. Section 3 describes the research approach and analysis procedures. Section 4 reports the results and discusses implications. Section 5 concludes with contributions, limitations, and directions for future research.

6.2. Related Work

Requirements Engineering (RE) is often regarded as a front-end activity in software systems development process, validating the feasibility of a project and the assessment of the associated risks [160]. RE consists out of different activities such as product planning (including vision, business case analysis, requirements triage, requirements prioritization [26], roadmapping and release planning), stakeholder negotiation (including communication and agreeing on them [26, 160]), requirements management, elicitation, analysis [26, 160, 200], analysis and checking [26, 201]. This study will focus on requirements prioritization (RP).

Requirements prioritization (RP), or task prioritization (TP) is considered an important decision-making activity in Requirements Engineering [145] and is considered the most researched area in software development [202] because it's one of its crucial activities [203-205]. Still, finding the best approach to RP is difficult [206], even with the wide range of available requirements prioritization methods (see Table 26) [53, 62, 151, 167, 185, 189, 203, 207-217] and criteria.

Table 28: Requirements prioritization methods (alphabetical order)

Method	Short description
Aggregated DM methods	Combine results from multiple decision-making (DM) techniques to achieve a consensus ranking of requirements.

Aggregated Randomization Indices Method (AIRM)	Uses randomization and aggregation of multiple ranking indices to reduce bias and improve prioritization robustness.
Analytic Hierarchy Process (AHP)	A structured multi-criteria method where requirements are pairwise compared to derive relative priorities based on consistency ratios.
Assumption Mapping	Visual method that prioritizes requirements or ideas by mapping assumptions against certainty and impact to focus validation efforts.
Binary Search Tree (BST)	A computational structure used to iteratively insert and compare requirements, yielding an ordered hierarchy based on priority.
Bubble Sort	Simple comparative sorting algorithm that repeatedly swaps adjacent requirements until they are ordered by priority.
Buy a Feature	Stakeholders are given budgets to “buy” preferred features, revealing relative value and collective preferences.
Case Based Ranking (CBR)	Extension of case-based reasoning that ranks requirements by comparing new cases with past successful projects.
Case Based Reasoning (CBR)	Prioritization approach that uses past similar projects (cases) to infer appropriate priorities for new requirements.
Cost / Value Approach	Prioritizes requirements by comparing estimated implementation cost against expected business or customer value.
Cumulative Voting (100-point method)	Stakeholders distribute 100 points across requirements to express perceived importance or value.
Data Envelopment Analysis (DEA)	Efficiency-based quantitative method treating each requirement as a decision unit to evaluate input/output ratios.
Dot Voting	Quick, participatory technique where team members place dots on preferred requirements to visualize collective priorities.
E ³ -value	Economic modeling approach that analyzes value exchanges among actors to prioritize requirements maximizing overall value creation.
Eclipse Framework	Tool-supported framework integrating different prioritization methods (e.g., AHP, weighting) within the Eclipse environment.
ELECTRE	Multi-criteria outranking method that compares alternatives pairwise to determine dominance based on concordance/discordance indices.
Expert or Management Opinion	Relies on domain experts or managers to subjectively rank requirements based on experience and intuition.
Feature Buckets	Groups requirements into thematic “buckets” (e.g., metrics, customer delight) and assigns relative investment priorities.
Frequency of Use	Prioritizes requirements according to how often a feature

		is expected to be used by end-users.
Fuzzy RDF FG		Fuzzy logic-based reasoning framework for requirements prioritization using RDF graphs and fuzzy inference.
Fuzzy Theory		Applies fuzzy logic to model uncertainty and imprecision in stakeholders' judgments when ranking requirements.
Hypothesis Canvas	Prioritization	Lean validation tool to rank hypotheses or assumptions underlying requirements by importance and evidence strength.
ICE (Impact, Confidence, Ease)		Scores ideas or requirements on impact, confidence, and ease to identify those with highest expected payoff.
Impact/Effort Matrix		Two-dimensional matrix classifying requirements by expected impact versus implementation effort.
Integrated Approach	Prioritization	Combines qualitative and quantitative prioritization methods to balance stakeholder input and analytical rigor.
Interactive Algorithm	Genetic	Evolutionary algorithm that iteratively evolves requirement sets based on stakeholder feedback and fitness functions.
Kano Model		Categorizes requirements into must-have, performance, and excitement attributes to guide prioritization by customer satisfaction.
Lanchester Theory		Applies competitive dynamics models (from military strategy) to prioritize requirements maximizing competitive advantage.
Minimal Spanning Tree		Graph-based method connecting requirements with minimal cost paths to identify dependencies and optimal sequencing.
MOSCOW		Classifies requirements into Must-have, Should-have, Could-have, and Won't-have categories.
Multi-attribute Theory (MAUT)	Utility	Evaluates each requirement based on multiple weighted attributes, aggregating them into a single utility score.
Multi-criteria Making (MCDM)	Decision-	General family of quantitative techniques that evaluate alternatives using several weighted criteria.
Numerical Assignment		Assigns numeric values to requirements according to predefined scales of importance or urgency.
Opportunity Scoring		Ranks features by contrasting customer importance with satisfaction to identify high-opportunity areas.
Planning Game		Agile technique where developers and customers collaboratively estimate and prioritize user stories for release planning.
Priority Groups		Requirements are placed into ordered groups (high, medium, low) instead of absolute ranks.
PROMETHEE		Outranking method that compares alternatives using preference functions to produce partial or complete rankings.
Quality Deployment (QFD)	Functional	Translates customer needs into engineering characteristics and prioritizes requirements based on weighted relationships.
Qualitative Win-Win		Negotiation-based approach that seeks solutions satisfying

	all stakeholders' key objectives.
Ranking based on Product Definition	Prioritizes requirements according to alignment with the agreed product vision or definition document.
RePizer	Tool-supported requirements prioritization method combining multiple techniques to generate ranked lists efficiently.
RICE (Reach, Impact, Confidence, Effort)	Quantifies and ranks requirements using reach, impact, confidence, and effort scores.
Round the Group Prioritization	Facilitated group technique where each participant successively adds or adjusts priorities in a round-robin fashion.
SAW (Simple Additive Weighting)	Linear multi-criteria method that computes a weighted sum of normalized scores for each requirement.
SMART (Simple Multi-Attribute Rating Technique)	Scores and weights attributes of requirements to calculate overall importance in a transparent way.
Story Mapping	Visual method that orders user stories along a user journey to reveal priorities and release slices.
Systemico Model	Systems-thinking-based model assessing interdependencies among requirements for holistic prioritization.
TOPSIS	Multi-criteria method that ranks alternatives by distance from an ideal and a negative-ideal solution.
Top 10 Requirements	Simple ranking approach listing the ten most important requirements based on stakeholder consensus.
Value Oriented Prioritization (VOP)	Focuses on maximizing overall business or customer value derived from implemented requirements.
Weighted Scoring	Assigns numerical weights to criteria and computes total weighted scores for each requirement.
Weighted Shortest Jobs First (WSJF)	Agile prioritization formula dividing value (or cost of delay) by job duration to maximize economic throughput.
Wiegiers Matrix Approach	Compares requirements pairwise on benefit, penalty, cost, and risk to compute a composite priority score.
\$100 Allocation	Stakeholders distribute \$100 among requirements to express relative importance or perceived value.

Besides the models, some studies have focused on choosing the requirements prioritization criteria, in the pursuit of improving the performance of predictors [172], both for functional as, non-functional [218] requirements prioritization efforts. This seldomly gets done considering a startup context. Specifically, Ma [219] and Hujainah, Bakar [53] emphasize the selection criteria and methods employed within requirements engineering (RE) while not addressing the startup context. On the other hand, Gupta, Fernandez-Crehuet [150] demonstrates a strong focus on the startup context, yet it remains descriptive and does not link back to the perspective of practitioners.

Regardless of the context and origin of the identified requirements prioritization criteria, they can be grouped across 10 categories as defined by Svahnberg, Gorschek [21].

6.3. Research Methodology

We employed a mixed-method approach to explore relevant requirements prioritization criteria

(RPC) in the context of software startups, incorporating both theoretical and practitioner perspectives. The theoretical perspective was examined through a systematic literature review (SLR), while the practitioner view was captured via 34 semi-structured interviews with 19 software startup founders, conducted during the first two years of their ventures. This combination of methods provides a comprehensive understanding of which RPCs are considered valuable in early-stage startups and highlights areas that warrant further research and practical attention.

6.3.1. Research Questions

The primary aim of this study is to comprehensively gather and synthesize relevant trace evidence concerning activities related to requirements prioritization criteria (RPC) in software startups. This is achieved through a systematic literature review, which explores existing academic research within the software startup domain, and through semi-structured interviews with startup founders, which offer practical insights into how these criteria are applied in real-world scenarios. Together, these complementary methods provide a well-rounded understanding of RPC in early-stage software ventures.

By examining the available literature, our investigation seeks not only to identify existing practices, contrast theory from reality, but also to structure and envision their impact on startups, considering current best practices. To guide this endeavor, a set of research questions (RQs) has been formulated and is presented in Table 27. Through this study, we aspire to offer valuable insights that can inform decision-making processes and contribute to the overall success and sustainability of software startups in today's dynamic business landscape.

Table 29: Research questions.

RQ	Research Question	Purpose
RQ1	What requirements prioritization criteria are reported in the academic literature on requirements engineering?	To establish a comprehensive catalogue of requirements prioritization criteria derived from existing scholarly research.
RQ1.1	What requirements prioritization criteria are reported in the academic literature on requirements engineering in a software startup context?	To establish a comprehensive catalogue of requirements prioritization criteria derived from existing startup focused scholarly research.
RQ2	Which requirements prioritization criteria are applied in software startups by practitioners?	To determine which of the literature-identified criteria are actually used in the startups under study and to identify any new or context-specific criteria that emerge from practice.
RQ3	How do financially oriented requirements prioritization criteria relate to indicators of startup performance and survival?	To explore the association between the use of financial prioritization criteria and observed performance tiers among startups.

The research questions were refined to improve their logical sequence and methodological alignment. RQ1 and RQ1.1 focus on identifying requirements prioritization criteria from the academic literature — first broadly, then specifically within a startup context. RQ2 examines

which criteria are actually applied in practice. RQ3 explores how financially oriented criteria relate to startup performance and survival.

The following research questions aim to systematically identify, analyze, and contextualize requirements prioritization criteria (RPC) relevant to software startups. They connect the theoretical insights from the structured literature review with empirical observations from multiple case studies. Together, these questions form the analytical backbone of this study and guide the synthesis presented in the subsequent sections.

RQ1. *What requirements prioritization criteria are reported in the academic literature on requirements engineering?* This question establishes the foundation of the study by identifying which criteria have been discussed in scholarly research on requirements prioritization. It supports the construction of a comprehensive catalogue of Requirements Prioritization Criteria (RPC) derived from existing studies and provides the theoretical baseline for comparison with empirical evidence from software startups.

RQ1.1. *What requirements prioritization criteria are reported in the academic literature on requirements engineering in a software startup context?* This sub-question narrows the scope of RQ1 by focusing specifically on startup-oriented scholarly research. It examines whether the criteria identified in the broader literature hold within a startup context or whether the startup-specific literature surfaces a distinct or more focused set of prioritization criteria, providing a more targeted theoretical baseline for the empirical work that follows.

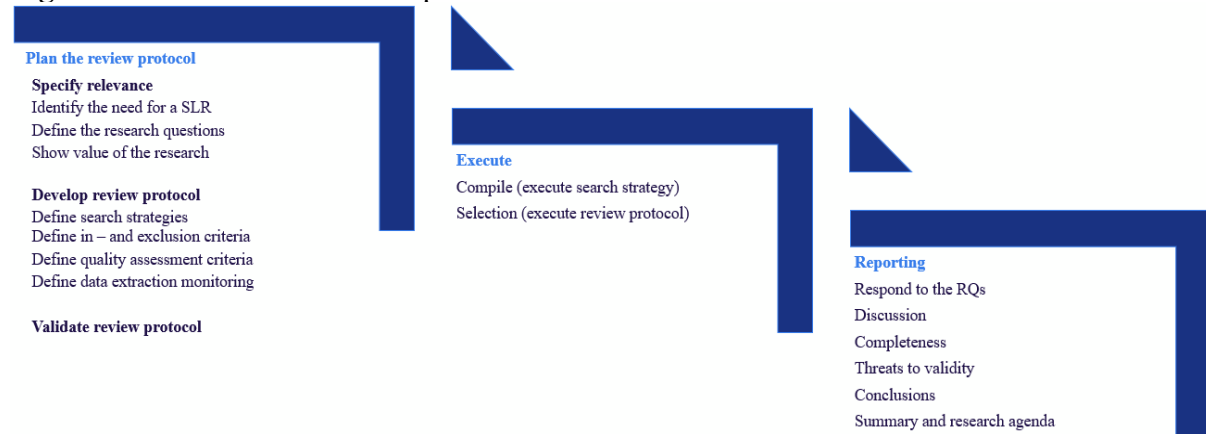
RQ2. *Which requirements prioritization criteria are applied in software startups by practitioners?* This question investigates which of the criteria identified in the literature are actually used in the startups under study and whether additional or context-specific criteria emerge from practice. It bridges the systematic literature review and the multiple-case study by connecting theory to observed reality.

RQ3. *How do financially oriented requirements prioritization criteria relate to indicators of startup performance and survival?* This final research question explores the relationship between the use of financially focused criteria — such as cost, value, and ROI — and the relative performance of startups. The analysis aims to identify whether financially oriented prioritization patterns correspond with stronger-performing cases, and whether such associations hold across different startup contexts and performance tiers.

6.3.2. Review Protocol Development

Our review protocol methodology (Figure 17) is an extended adaptation of the structures described by Kitchenham [220], Brereton, Kitchenham [221] and Kitchenham, Brereton [61], consequently reduces the possibility of researcher bias [220]. When it comes to the first step of planning, identifying the necessity for a SLR (see Section 2.1), defining the research questions and their value (see Section 3.1) have already been discussed.

Figure 17: Overview of review protocol



The final goal the defined search strategy is compiling a master list of studies that will be considered to be part of the structured literature review. The unfiltered search results should be saved and retained for possible reanalysis. For this structured literature review, only relevant primary studies will be considered (journal papers and conference proceedings). This means that publications from book, research registers, internet (blog posts, website...) and professional journals get excluded.

To comprehensively search for related studies, our search strategy began with an online search of digital libraries. Studies that are related to this review were extracted from four relevant digital libraries and databases for conducting a SLR [222], namely, IEEE Xplore, ScienceDirect, Springer Link and ACM Digital Library, using the following search terms: ((Criteria OR aspects OR attributes OR variables OR category OR variable OR categories) AND (Requirements OR Requirement) AND (prioritization OR selection OR triage)).

Required to limit the research domain because there's already written a lot on the topic [15]; it is advisable to limit and structure the presentation. The following inclusion (Table 28), exclusion (Table 29) and quality assessment criteria (Table 30) have been considered for validation.

Hujainah, Bakar [53] opted to only include papers that reach a score of 50% for the inclusion criteria - our benchmark before a pilot study will be the same. This means setting the bar at a score of 2.5 out of 5. For the exclusion criteria; when the sum is less than 2, the paper gets excluded.

The main goal of quality assessment criteria is to increase the confidence level that the study is actually relevant and methodologically solid. This in order to make sure that the outcome of the study makes it possible to guide future research recommendations. Only the papers that have survived the previous hurdle (the inclusion and exclusion criteria) are in scope to have their quality assessed. In software engineering, usually all levels of evidence get accepted.

To ensure methodological rigor, only peer-reviewed academic publications were included in the review. Non-academic sources such as blogs, trade magazines, or books without peer review were excluded. As all retained studies met these quality criteria, no additional exclusion thresholds or evidence-level classifications were applied.

Table 30: Inclusion criteria

IC	Inclusion criteria	1	0.5	0
1	Reports on requirements prioritization criteria	In-depth (multiple criteria)	Basic (single criterium)	None
2	Reports on context	Startup	General	None

3	Reports results of the prioritization efforts	Explicitly mentioned	Basic	None
4	Paper within the Requirements Engineering domain	Explicitly mentioned	Linked	No
5	Paper related to software products	Software	General	Other

Table 31: Exclusion criteria

	Exclusion criteria	1	0.5	0
1	Year of publication before 1983	≥ 1983	N/A	Before 1983
2	There is no PDF file available	PDF available	N/A	No available

Table 32: Quality assessment criteria

	Quality assessment criteria	2	1	0
QAC1	Citations	Citations ≥ 100 (top 2%)	C between 100 AND 5	C < 5 (lower 50%)
QAC2	Number of references	#ref ≥ 25	#ref between 25 AND 10	#ref < 10
QAC3	Is the proposed methodology (criteria) clearly explained?	Yes	Moderate	No
QAC4	Is the evaluation of proposed criteria on adequate case studies, subjects or project data sets?	Subjects >10	Subjects between 10 AND 4	Subjects <4
QAC5	Is the result of the study clearly stated?	Yes	Moderate	No

QAC1: Citations. According to Scott [223] regardless the year of the publication, having over 100 citations can be considered impactful, while having fewer than five makes the paper part of the lower half. This is also a tactic used by Krishnan and Ulrich [5]. The number of citations is based on Google scholar, or Research Gate depending on availability of the data.

QAC2: References. Ucar, López-Fernandino [224] sees an increase in the number of references in academic papers, with median of 25 references per engineering journal papers in 2013.

QAC3: Proposed methodology. To insure that the quality of the paper meets scientific rigor, it's a must to include a criteria that evaluates the used methodology (technique or solution) of the study [53].

QAC4: Reference data. This criterion assesses whether the reported results are supported by an adequate empirical basis, such as the number of analyzed cases, datasets, or experiments. Studies with an extremely limited empirical foundation may provide valuable qualitative insights but are less suitable for drawing generalizable conclusions. Therefore, the scope of underlying data is considered as one factor contributing to the overall credibility of the study's findings.

QAC5: Stated result. When the methodology and the research data are at par, the final element regarding to study structure that deserves some additional scrutiny is how clearly the study results are stated [53, 225]. Although it's not a commonly supported practice to use basic arithmetic to decide whether or not the quality of a paper is good enough, it does get done by Hujainah, Bakar [53] and Achimugu, Selamat [62]. So here the maximum score that's possible

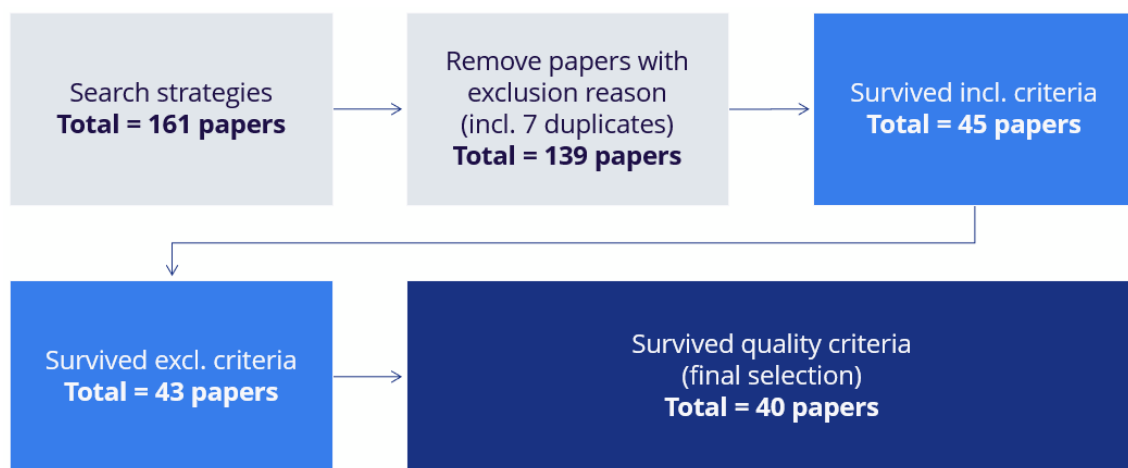
is 5 times 2, making 10. Papers scoring below 5 get excluded from the study.

Validate Review Protocol. The review protocol was piloted on an initial subset of one hundred studies identified through the search strategy [49, 226]. This pilot test served to verify that the inclusion, exclusion, and quality assessment criteria could be consistently applied and that the search strings retrieved relevant studies. Minor refinements were made to the criteria before executing the full review.

Execute Research Strategy

Figure 18 shows that the search strategy resulted in a total of 161 papers, of which 40 have survived the review protocol. Considering the limit number of initial studies, an acceptance rate of 25% is highly recommended [5, 53, 62, 225]. Considering the data, this thresholds gets basically achieved with an positive acceptance rate of 24.84%.

Figure 18: Protocol result search strategy



6.3.3. Semi-structured Interviews

We conducted 34 interviews with the founders of 19 software startups (see Table 31), using a purposive sampling method [12, 71]. All software startups were part of the same quarterly cohort (Q4/2022), of the same startup accelerator in Belgium. To ensure homogeneity as much as possible, three startups (S3, S6 and S19) got excluded because they were already too long in business, or were a non-profit. In line with common definitions in startup research [27, 30], companies older than five years were no longer considered startups. The first interview took place in Q4 of 2022 and the final interview, including data gathering in Q2 of 2024. We followed the general interview guidelines Patton [139] to ensure consistency and reliability in our data collection. In accordance with our ethics policy, we ensured that all participants and their shared information would remain confidential. We performed thematic analysis [140] on the transcripts of the interviews. The guiding questions that have been asked related to the study were; (1) How do you determine, due to resource constraints, which ones make the cut and which ones don't? (2) For those that make it, how do you prioritize them? (3) Which requirements prioritization criteria do you use? and (4) Which prioritization methods do you use?

Table 33: Startups and interviews

ID	Role	Sector	Interviews	Years in business	For profit?
S1	Co-founder & CTO	Football data analytics	1 (1h15)	3	Yes
S2	Co-founder & CTO	Security platform	2 (2h07)	2	Yes
S3	CEO	AI for mobile mapping	2 (0h57)	7	Yes
S4	Founder & Product Manager	Web3 SocialFi	3 (2h45)	3	Yes
S5	CEO & co-founder	PLG platform	2 (1h13)	4	Yes
S6	Co-owner	Legal tech	2 (2h35)	5	Yes
S7	Co-founder	HR tech	1 (0h45)	2	Yes
S8	Founder	Logistical solution	1 (1h25)	3	Yes
S9	Co-founder	Form builder	3 (3h24)	3	Yes
S10	Co-founder & CTO	Web2 credit market	2 (1h45)	3	Yes
S11	CEO	HR tech	3 (2h53)	3	Yes
S12	CEO & founder	Healthcare platform	2 (1h48)	2	Yes
S13	Co-owner	HR tech	2 (1h15)	3	Yes
S14	CEO & founder	HR tech	1 (2h11)	3	Yes
S15	CEO & co-founder	Data document management	3 (2h41)	3	Yes
S16	Founder	Fashion tech	1 (0h58)	2	Yes
S17	CEO & co-founder	AI for trends analysis	1 (1h23)	4	Yes
S18	Co-founder	HR tech	1 (0h51)	4	Yes
S19	Co-founder	HR-tech	1 (1h34)	2	No

6.3.4. Coding

The coding process followed two sequential phases. First, all criteria explicitly mentioned in the context of requirements prioritization were extracted verbatim from the 50 primary sources, resulting in 502 criterion citations. Synonymous and semantically overlapping labels were then normalized into unique entries, reducing the list to 82 unique criteria, which were subsequently mapped onto the 10 overarching categories identified in the literature. Second, the same 82 criteria served as a coding frame for the interview data. Interview transcripts were coded deductively against this frame, with each prioritization-related statement mapped to its corresponding criterion where possible. Statements that could not be mapped to any existing criterion were flagged separately.

6.4. Results

RQ1: What requirements prioritization criteria are reported in the academic literature on requirements engineering?

Of collated studies, two thirds are journal publications. When drilling down on the journals, Information and Software Technology takes the number one position with 21.05% of all related publications, followed by the Journal of Systems and Software, good for 10.53%. The rest of them only represent 5% or less. When taking a closer look to the conference publications, the largest source (20%) can be traced back to IEEE. Besides the IEEE conferences, all the rest are one-offs.

The examination of the temporal distribution of these publications related (Figure 19) to requirements prioritization criteria activities reveals interesting peaks. On average, there are two publications per year. However, two standout years, 2003 and 2010, demonstrate exceptional activity with both 5 publications each. Notably 3 out of 4 studies during the 2018-2019 period have been observed to consider a startup context. Even more interestingly, these are 100% of the retained publications related to that field of inquiry. This surge in research aligns with a period of prosperity and success for startups, which is not surprising given the increasing attention and significance attributed to the startup ecosystem during that time. This observation highlights the growing interest in understanding and addressing the unique challenges and opportunities faced by product managers within the dynamic and thriving environment of startups.

Beyond publication patterns, the empirical depth of the included studies also varies substantially. Most papers within this field of research rely on a very limited number of case studies—typically fewer than five [3, 17, 227, 228]—a limitation often attributed to confidentiality constraints when studying live startups [17, 195]. A smaller subset of studies includes between five and ten cases [11, 178, 229, 230], while only a handful exceed ten [26, 173, 231]. According to Barney, Aurum [3], such small sample sizes reduce the statistical generalizability of findings, although they may still provide valuable qualitative insights. This overall trend underscores the methodological challenges of accessing sufficient empirical data in startup-oriented requirements engineering research.

Figure 19: Number of papers by year of publication

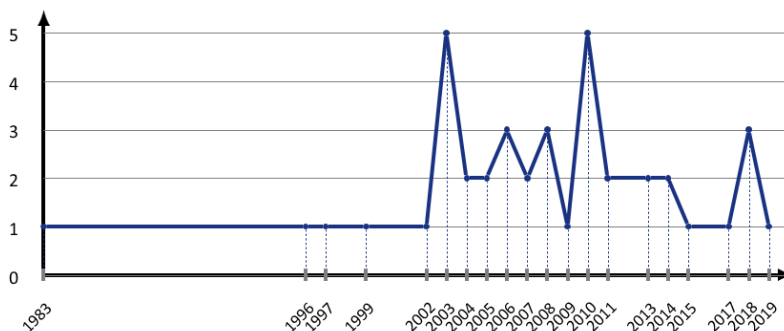


Table 31 presents the top ten most frequently mentioned Requirements Prioritization Criteria (RPC), together with their relative frequencies and associated reference studies. The full list of 82 identified criteria and their corresponding sources is available in Table 32. Notably, the activity garnering the most significant attention is the expected cost (including development), accounting for 12.75% of the discussions. However, the most intriguing observation arises from the fact that close to three quarters (59/82) of them are referenced merely up to 1 percent, indicating considerable untapped potential for further and in-depth scholarly exploration.

Table 34: Overview of distribution of identified requirements prioritization criteria (top 10)

Requirements Prioritization Criteria	% of all citations
Expected cost (incl. development)	12.75%
Risks (incl. dev, business, failure...)	6.18%
Resource constraints (incl. HR)	5.98%
Dependencies (incl. requirements, customers, technical...)	5.78%
Time constraints	4.98%
Stakeholder preferences	4.58%
Business value	4.18%
Customer value	2.79%
Value (generic)	2.79%
Estimated expected revenue	2.79%

Table 32 shows in detail which RPC that got discussed in each of the retained studies. For each of these studies, the full paper has been thoroughly read and coded in order to identify all mentioned RPC.

Table 35: Selected studies and their RP criteria

Ref.	Cited RP criteria
[211]	Profitability
[174]	Predictability (incl. commercial, info), Return on investment (ROI), Risks (incl. dev, business, failure...)
[176]	Complexity, expected cost (incl. Development), Time to market
[166]	Business value, Change management, Dependencies (incl. requirements, customers, technical...), Expected cost (incl. development), MVP feature set coverage, Resource constraints (incl. HR), Risks (incl. dev, business, failure...)
[165]	Expected cost (incl. development), Financial value, Value (generic)
[168]	Budget constraints, Risks (incl. dev, business, failure...), Time constraints, Time to market
[164]	Customer value, Expected cost (incl. development), Risks (incl. dev, business, failure...)
[167]	Expected cost (incl. development), Feasibility, Quality of service, Time constraints
[232]	Business value
[173]	Business value, Expected cost (incl. development), Return on investment (ROI), Stakeholder preferences, Value (generic)
[11]	Cost benefit ratio, Expected cost (incl. development), Maintenance, Resource constraints (incl. HR), Time constraints
[157]	Cultural alignment
[26]	Cost benefit ratio, Customer value, Dependencies (incl. requirements, customers, technical...), Expected cost (incl. development), MVP feature set coverage, Need to demonstrate product, Prioritization (incl. features, bugs), Stakeholder preferences, Strategy alignment
[31]	Business value, Customer value, Dependencies (incl. requirements, customers, technical...), Easy to use, Efficiency (incl. decreasing costs), Estimated expected revenue, Expected cost (incl. development), Prioritization (incl. features, bugs), Profitability, Quality of service, Reduce errors, Resource constraints (incl. HR), Time to market, User engagement
[125]	Customer value

[233]	Budget constraints, Quality of service, Time constraints, Time to market
[187]	Dependencies (incl. requirements, customers, technical...), Expected cost (incl. development), Importance, Penalties, Risks (incl. dev, business, failure...), Time constraints, Volatility
[178]	Budget constraints, Cost benefit ratio, Customer value, Dependencies (incl. requirements, customers, technical...), Expected cost (incl. development), Risks (incl. dev, business, failure...), Time constraints
[169]	Cost importance ratio, Expected cost (incl. development), Importance
[209]	Financial value, Resource constraints (incl. HR), Stakeholder preferences, Strategic value, Time constraints
[190]	Dependencies (incl. requirements, customers, technical...), Expected cost (incl. development), Importance, Stakeholder preferences
[184]	Competitive intensity, Customer value, Estimated market share, Expected cost (incl. development), Financial value, Penalties, Risks (incl. dev, business, failure...), Time constraints, User engagement, Value (generic)
[62]	Expected cost (incl. development), Quality of service, Resource constraints (incl. HR), Time constraints
[54]	Business value, Customer value, Dependencies (incl. requirements, customers, technical...), Expected cost (incl. development), Penalties, Risks (incl. dev, business, failure...), Time constraints, Value (generic)
[234]	Customer value, Dependencies (incl. requirements, customers, technical...), Expected cost (incl. development), Prioritization (incl. features, bugs)
[213]	Customer value, Resource constraints (incl. HR), Technical constraints, Time constraints
[179]	Documentation quality, Efficiency (incl. decreasing costs), Estimated expected revenue, Expected cost (incl. development), Reduce errors, Time constraints
[186]	Budget constraints, Competitive intensity, Customer value, Expected cost (incl. development), Regulations, Time constraints, User satisfaction
[53]	Budget constraints, Business value, Complexity, Consistency, Customer value, Dependencies (incl. requirements, customers, technical...), Easy to use, Estimated expected revenue, Expected cost (incl. development), Feasibility, Importance, Level of ambiguity, Maintenance, Modifiability, Penalties, Predictability (incl. commercial, info), Prioritization (incl. features, bugs), Quality of service, Redundancy (incl. data), Reliability, Resource constraints (incl. HR), Re-usability, Risks (incl. dev, business, failure...), Security, Stakeholder preferences, Technical constraints, Technology, Testability, Time constraints, Traceability, User satisfaction, Value (generic), Volatility
[126]	Competitive intensity, Cost of taken no action, Dependencies (incl. requirements, customers, technical...), Expected cost (incl. development), Financial value, MVP feature set coverage, Predictability (incl. commercial, info), Regulations, Resource constraints (incl. HR), Signed contracts, Stakeholder preferences, Strategy alignment, Time to market
[3]	Change management, Competitive intensity, Complexity, Cost benefit ratio, Dependencies (incl. requirements, customers, technical...), Estimated expected revenue, Feature promised, Maintenance, Profitability, Resource constraints (incl. HR), Signed contracts, Stakeholder preferences, Time to market, Volatility
[10]	Company synergies, Competitive intensity, Complexity, Durability, Existence of market need, Market maturity (incl. growth rate), Payback period, Predictability (incl. commercial, info), Profitability, Proprietary position, Regulations, Resource

	constraints (incl. HR), Strategic value, Strategy alignment, Time to market
[18]	Estimated expected revenue, Expected cost (incl. development), Partnership, Return on investment (ROI), Stakeholder preferences
[79]	Business value, Customer value, Expected cost (incl. development), Non-functional characteristics, Strategy alignment
[19]	Prioritization (incl. features, bugs)
[235]	Return on investment (ROI)
[9]	Budget process, Estimated expected revenue, Financial value, Internal rate of return (IRR), Payback period, Profitability, Return on investment (ROI)
[155]	Cash flow analysis, Estimated expected revenue, Estimated market share, Expected cost (incl. development), Financial value, Profitability
[172]	Break-even point
[21]	Budget constraints, Business value, Complexity, Customer value, Dependencies (incl. requirements, customers, technical...), Estimated expected revenue, Expected cost (incl. development), Financial value, Penalties, Prioritization (incl. features, bugs), Quality of service, Resource constraints (incl. HR), Risks (incl. dev, business, failure...), Stakeholder preferences, Technical constraints, Time constraints, Time to market, Value (generic), Volatility

Table 33 gives the overview of all identified criteria, allocated to one of the categories. In the end, this resulted in 82 final selection criteria across 10 categories as defined by Svahnberg, Gorschek [21]. Here it gets confirmed that the category of cost-related factors, with 18.73% is the one that's most discussed.

Table 36: Overview of the RPC across categories

Category	Criteria	Count cited
Cost-Related Factors: includes criteria related to budget, cost constraints, cost-benefit analysis, costs of not implementing a requirement, development costs, and cost of maintenance.	Expected cost (incl. development) Budget constraints Cost benefit ratio Efficiency (incl. decreasing costs) Maintenance Budget process Cost of taken no action Cost importance ratio	94 (18.73%)
Revenue and Financial Impact: covers criteria such as revenue generation, financial benefits, profitability, return on investment (ROI), and net present value.	Estimated expected revenue Value (generic) Profitability Financial value Return on investment (ROI) Payback period Net present value (NPV) Break-even point Cash flow analysis Internal rate of return (IRR)	57 (11.35%)
Business and Strategic Impact: encapsulates criteria related to	Business value Estimated market share	58 (11.55%)

business goals, strategic benefits, market share estimates, competitive differentiation, and alignment with product strategy.	Competitive intensity Strategy alignment MVP feature set coverage Strategic value Signed contracts Market maturity (incl. growth rate) Proprietary position Cultural alignment Brand value Company synergies Need to demonstrate product Existence of market need Feature promised	
Stakeholder and Customer Considerations: comprises criteria related to customer needs, customer value, stakeholder value, stakeholder preferences, and customer satisfaction.	Stakeholder preferences Customer value Importance User satisfaction Change management User engagement Contribution to user task User loyalty Frequency of use Communication Partnership	60 (11.95%)
Technical Factors and Constraints: includes criteria related to technical feasibility, complexity of development, technology constraints, product technology, and testability	Complexity Technical constraints Code change Re-usability Modifiability Redundancy (incl. data) Technology Compatibility (incl. technical) Scalability Standards	27 (5.38%)
Resource Constraints and Availability: covers criteria such as resource constraints, availability of human resources, skilled personnel, and other re-sources.	Resource constraints (incl. HR)	30 (5.89%)
Quality and Performance Metrics: encompasses criteria like quality of service, performance, reliability, maintainability, and error rates.	Quality of service Easy to use Documentation quality Level of ambiguity User experience Reduce errors Reliability Consistency Durability	35 (6.97%)

	Traceability Testability Non-functional characteristics	
Risk Assessment: incorporates criteria related to risks, including development risks, business risks, risk of failure, and risk of implementing a feature.	Risks (incl. dev, business, failure...) Penalties Prioritization (incl. features, bugs) Predictability (incl. commercial, info) Volatility Security Feasibility Regulations Legal mandate Harm avoidance Issues	74 (14.74%)
Time Constraints and Scheduling: includes criteria related to development time, time to market, delivery time, and scheduling constraints.	Time constraints Time to market	37 (7.37%)
Dependency and Interdependency Factors: Covers criteria such as dependencies between requirements, feature dependencies, and requirement precedence constraints.	Dependencies (incl. requirements, customers, technical...) Similarity	30 (5.98%)
		= 502 (100%)

RQ1.1. Which requirements prioritization criteria are applied in software startups?

The examination of requirements prioritization criteria within the startup domain reveals a notable scarcity of research. Out of 40 analyzed studies, 12 of them give some context about the companies that were involved, only 3 explicitly focus on a software startup context (Table 34). Overall, this is a clear indication that the startup context, and the related RPC, is of relative under-researched nature.

Table 37: Studies considering a software startup context

Ref.	Title	Startup?
[166]	Agile requirements prioritization in large-scale outsourced system projects: An empirical study	
[173]	Prioritization of quality requirements: State of practice in eleven companies	
[26]	A model of requirements engineering in software startups	X
[31]	An anatomy of requirements engineering in software startups using multi-vocal literature and case survey	X

[125]	Towards prioritizing software business requirements in startups	X
[187]	Challenges and future trends in software requirements prioritization	
[178]	Do we know enough about requirements prioritization in agile projects: insights from a case study	
[169]	Supporting the selection of software requirements	
[190]	A comparison of nine basic techniques for requirements prioritization	
[62]	A systematic literature review of software requirements prioritization research	
[54]	Value-oriented requirements prioritization in a small development organization	
[3]	A product management challenge: Creating software product value through requirements selection	

In contrast to the 82 RP criteria, when considering all references, there are only 19 (Table 35) to be found when explicitly taking the startup context into account, across 3 studies [26, 31, 125].

Table 38: Requirements prioritization criteria in a startup context

Category	Criteria	Criteria (startup context)
Cost-Related Factors	Expected cost (incl. development)	x
	Budget constraints	
	Cost benefit ratio	x
	Efficiency (incl. decreasing costs)	
	Maintenance	
	Budget process	
	Cost of taken no action	
	Cost importance ratio	
Revenue and Financial Impact	Estimated expected revenue	
	Value (generic)	
	Profitability	
	Financial value	
	Return on investment (ROI)	
	Payback period	
	Net present value (NPV)	
	Break-even point	
	Cash flow analysis	
	Internal rate of return (IRR)	

Business and Strategic Impact	Business value	x
	Estimated market share	
	Competitive intensity	
	Strategy alignment	x
	MVP feature set coverage	x
	Strategic value	x
	Signed contracts	
	Market maturity (incl. growth rate)	
	Proprietary position	
	Cultural alignment	
	Brand value	
	Company synergies	
	Need to demonstrate product	x
	Existence of market need	
	Feature promised	
Stakeholder and Customer Considerations	Stakeholder preferences	
	Customer value	x
	Importance	x
	User satisfaction	
	Change management	
	User engagement	
	Contribution to user task	x
	User loyalty	
	Frequency of use	x
	Communication	
	Partnership	
Technical Factors and Constraints	Complexity	x
	Technical constraints	
	Code change	
	Re-usability	
	Modifiability	
	Redundancy (incl. data)	
	Technology	
	Compatibility (incl. technical)	
	Scalability	
	Standards	
Resource Constraints and	Resource constraints (incl. HR)	x

Availability		
Quality and Performance Metrics	Quality of service	
	Easy to use	
	Documentation quality	
	Level of ambiguity	
	User experience	
	Reduce errors	
	Reliability	
	Consistency	
	Durability	
	Traceability	
	Testability	
	Non-functional characteristics	
Risk Assessment	Risks (incl. dev, business, failure...)	x
	Penalties	
	Prioritization (incl. features, bugs)	x
	Predictability (incl. commercial, info)	
	Volatility	
	Security	
	Feasibility	x
	Regulations	
	Legal mandate	
	Harm avoidance	
	Issues	
Time Constraints and Scheduling	Time constraints	x
	Time to market	x
Dependency and Interdependency Factors	Dependencies (incl. requirements, customers, technical...)	x
	Similarity	
	= 82	= 19

When grouping on the category level it's clear that first cost-related factors is the number one interest (23.08%), and secondary more subjective interests that are getting considered as startup RPC with stakeholder and customer considerations (19.23%) and business and strategic impact (15.38%). When it comes to revenue and financial impact category, it's barely considered (Table 36).

Table 39: Overview of RPC across categories in a software startup context

Startup Requirements Prioritization Categories	% of all citations
Cost-Related Factors	23.08%
Stakeholder and Customer Considerations	19.23%
Business and Strategic Impact	15.38%
Quality and Performance Metrics	11.54%
Dependency and Interdependency Factors	7.69%
Risk Assessment	7.69%
Revenue and Financial Impact	7.69%
Time Constraints and Scheduling	3.85%
Resource Constraints and Availability	3.85%

RQ2. Which requirements prioritization criteria are applied in software startups by practitioners?

When mapping all criteria according literature against what practitioners use, we see that only a limited number of criteria is in fact considered (Table 37).

Table 40: Overview of used RPC by software startup practitioners vs literature

Category	Criteria	Criteria (startup context)	Criteria (practitioners)
Cost-Related Factors	Expected cost (incl. development)	x	x
	Budget constraints		
	Cost benefit ratio	x	
	Efficiency (incl. decreasing costs)		
	Maintenance		
	Budget process		
	Cost of taken no action		
	Cost importance ratio		
Revenue and Financial Impact	Estimated expected revenue		x
	Value (generic)		
	Profitability		
	Financial value		x
	Return on investment (ROI)		x
	Payback period		
	Net present value (NPV)		
	Break-even point		
	Cash flow analysis		
	Internal rate of return (IRR)		
Business and Strategic Impact	Business value	x	
	Estimated market share		

	Competitive intensity		
	Strategy alignment	x	x
	MVP feature set coverage	x	
	Strategic value	x	
	Signed contracts		x
	Market maturity (incl. growth rate)		
	Proprietary position		
	Cultural alignment		
	Brand value		
	Company synergies		
	Need to demonstrate product	x	
	Existence of market need		
	Feature promised		
Stakeholder and Customer Considerations	Stakeholder preferences		x
	Customer value	x	x
	Importance	x	
	User satisfaction		
	Change management		
	User engagement		
	Contribution to user task	x	
	User loyalty		
	Frequency of use	x	x
	Communication		
	Partnership		
Technical Factors and Constraints	Complexity	x	x
	Technical constraints		
	Code change		
	Re-usability		
	Modifiability		
	Redundancy (incl. data)		
	Technology		
	Compatibility (incl. technical)		
	Scalability		
	Standards		
Resource Constraints and Availability	Resource constraints (incl. HR)	x	
Quality and	Quality of service		

Performance Metrics	Easy to use		
	Documentation quality		
	Level of ambiguity		
	User experience		
	Reduce errors		
	Reliability		
	Consistency		
	Durability		
	Traceability		
	Testability		
	Non-functional characteristics		
Risk Assessment	Risks (incl. dev, business, failure...)	x	x
	Penalties		
	Prioritization (incl. features, bugs)	x	
	Predictability (incl. commercial, info)		
	Volatility		
	Security		
	Feasibility	x	
	Regulations		
	Legal mandate		
	Harm avoidance		
	Issues		
Time Constraints and Scheduling	Time constraints	x	x
	Time to market	x	x
	Time to Value		x
Dependency and Interdependency Factors	Dependencies (incl. requirements, customers, technical...)	x	x
	Similarity		
	= 82	= 19	= 15

Table 41: Overview of used RPC across categories by software startups vs scholarly investigations

Requirements Categories	Prioritization Criteria	Startup id	In SLR?
Cost-Related Factors	Expected cost (including development days)	S1, S2, S9, S10, S11, S12, S15, S16 and S17	Yes

Business and Strategic Impact	Strategic alignment	S4, S5, S14, S16 and S17	Yes
Time Constraints and Scheduling	Time to Value (TtV)	S2 and S8	No
Time Constraints and Scheduling	Time to Market (TtM)	S1, S2, S4, S8, S9 and S10	Yes
Revenue and Financial Impact	Business case (including expected revenue and ROI)	S2, S9, S11, S12, S15, and S17	Yes
Stakeholder and Customer Considerations	Stakeholder preferences (informal user discussions)	S8, S9 S12, S14 and S17	Yes
Stakeholder and Customer Considerations	Stakeholder preferences (count of user requests)	S8, S9, S10, S11, S14, S15 and S17	Yes

Table 37 shows that only 15 unique criteria have been identified, used by practitioners. There's only 14 out of 82 overlap with the broad requirements prioritization criteria literature and 9 out of 14 with the startup context specific literature. The most frequently applied criteria across startups were expected development cost and stakeholder preferences — both cited by the majority of participants — followed by customer value, strategy alignment, and estimated expected revenue. Table 38 shows the overview on category level. When we look more closely, this means that there's a single criterion mentioned by practitioners by not described in literature: Time to Value.

When looking at the criteria within the revenue and financial impact category, having a clear and measurable business case—such as expected ROI, probability to upsell customers (S12), contribution to a 1M ARR target (S15), 100% revenue certainty (S2, S8, S9, S11, S17), or the percentage of users expected to adopt a feature (S2, S15)—emerges as one of the most valued considerations among the most successful startups. These teams consistently prioritize features that are tied to financial outcomes and that can be tracked post-release, setting them apart from less successful startups, who often lack this structured approach.

For instance, startup S3 initially emphasized financial impact in their prioritization and demonstrated strong performance early in the study. However, by the end, they had moved away from this criterion—citing the effort required and frequent mismatches with reality—which coincided with a decline in their performance. Similarly, S9 reflected on their experience and shared that, if given the chance to start over, they would be far more disciplined in only building features backed by a strong, revenue-driven business case. Strikingly, none of the startups that ended up bankrupt had included financial impact as a prioritization criterion, highlighting the significant risks of overlooking this factor in early-stage product decisions.

The data suggests that neglecting to consider development costs during prioritization can have severe consequences—specifically, all startups in the study that did not factor in this criterion (S5, S7, and S15) were no longer operational at the time of last interview.

Interestingly, all interview-identified criteria except one—“Time to Value” (TtV)—were also

present in the academic literature summarized in Table 38 Based on founder insights, "Time to Value" refers to the period between inception, release of a feature and the moment when its intended business value—such as revenue, ROI, growth, exposure, or another measurable goal—is largely realized. This criterion, although absent in the literature sample, appears to be of practical significance in early-stage startup environments, suggesting a potential gap in academic coverage and an opportunity for further research.

RQ3. How do financially oriented requirements prioritization criteria relate to indicators of startup performance and survival?

Table 39 summarizes the distribution of the sixteen analyzed startups across four performance tiers. The classification was based on quantitative indicators—including the number of funding rounds (1), total funding raised (2), latest annual revenue (3) and profit (4), remaining runway (5), and operational status (6) —which together provided a comparative ranking within the cohort.

Table 42: Startups & performance

ID	(1)	(2)	(3)	(4)	(5)	(6)	Final Ranking
S2	1	2	3	4		Y	2.5
S10	1	1	1	11	2	Y	3.2
S11	3	5	2	3	5	Y	3.6
S18	3	13	4	1	1	Y	4.4
S15	2	4	5	9	4	Y	4.8
S12	3	9	7	1	5	Y	
S17	3	3	9	10	3	Y	5.6
S4	2	6	10	5		Y	5.7
S14	3	7	6	6	11	N	6.6
S5	3	8	10	8	6	N	7.0
S9	3	11	9	5	7	Y	7.0
S1	3	10	10	7	8	Y	7.6
S8	4	14	9	2	10	Y	7.8
S13	4	14	8	4	11	Y	8.2
S16	3	12	10	9		Y	8.5
S7	4	14	10		12	N	10

Figure 20 visualizes the relationship between the use of financially oriented Requirements Prioritization Criteria (RPC) and the performance levels of the participating startups. The results reveal a clear trend: startups that systematically applied financial criteria—such as expected cost, business case, and revenue impact—tended to achieve higher performance rankings. In contrast, startups that neglected these considerations were overrepresented in the lower performance tiers or had ceased operations by the end of the study. This pattern suggests that integrating financial reasoning into prioritization decisions contributes positively to startup resilience and survival prospects.

Figure 20: Use of financially related requirements prioritization criteria versus ranking



Table 39 complements these findings by summarizing the financial RPCs identified in the literature. While traditional financial metrics such as Return on Investment (ROI), Payback Period, and Net Present Value (NPV) appear in a limited number of studies [9, 18, 126, 155, 173, 174, 235], their use in startup-focused research remains rare. In fact, none of the reviewed studies addressing software startups explicitly employed these advanced financial ratios. Instead, the literature typically refers to broader cost- or value-related factors without quantitative evaluation.

Table 43: Financial RPC

Paper	Financial ratio
Boehm [176]	Return on investment (ROI)
Svensson, Gorschek [173]	Return on investment (ROI)
Cleland-Huang and Denne [156]	Net present value (NPV)
Cooper [10]	Payback period
Bekkers, van de Weerd [18]	Return on investment (ROI)
Fogelström, Barney [235]	Return on investment (ROI)
Gorchels [9]	Return on investment (ROI), Internal rate of return (IRR), Payback period
Cosse and Swan [155]	Cash flow analysis
Guyon and Elisseff [172]	Break-even point

Taken together, these findings highlight a gap between academic emphasis and entrepreneurial practice. Financially disciplined startups—those that explicitly evaluate ROI or time-to-value—demonstrated stronger performance outcomes, yet such criteria remain underrepresented in existing research. Bridging this gap presents a valuable opportunity for future studies to develop and validate financially grounded prioritization approaches tailored to early-stage, resource-constrained ventures.

6.5. Discussion

This section discusses the study’s findings in relation to each research question. It integrates

insights from the structured literature review and the multiple-case study to provide a consolidated understanding of how requirements prioritization criteria (RPC) are identified, applied, and interpreted in software startups.

RQ1 – What requirements prioritization criteria are reported in the academic literature on requirements engineering?

The review confirmed that the literature on requirements prioritization is extensive but highly fragmented. A total of 82 unique criteria were identified, yet most appeared in only one or two studies, suggesting a lack of convergence around a stable or widely accepted set of criteria. Thematically, these criteria span cost, value, risk, stakeholder influence, customer satisfaction, and technical feasibility. This variety illustrates the multidimensional nature of prioritization but also exposes how inconsistent and context-dependent existing frameworks remain.

Financial or economically oriented criteria (e.g., cost, value, ROI) are mentioned, but only in a minority of publications and typically without explicit operationalization. Most studies treat cost as a resource proxy rather than a true financial measure, reflecting a focus on effort estimation instead of return. Overall, the academic body of work provides a broad but uneven landscape of prioritization criteria—rich in conceptual diversity, yet thin in empirical consistency.

RQ2.1 – Which requirements prioritization criteria are applied in software startups?

In practice, software startups use a narrower but more pragmatic subset of criteria. Founders frequently emphasized expected cost, time constraints, and strategic alignment, aligning partially with the literature. However, a striking observation is the emergence of context-specific criteria that are rarely discussed in academic work. Most notably, Time to Value (TtV)—the time between feature release and realized business value—emerged across multiple cases as a key decision driver.

While cost and time-to-market remain important, founders consistently evaluated features based on how quickly they would contribute to measurable outcomes such as revenue, growth, or retention. This highlights that startups translate abstract prioritization concepts into direct survival metrics. The discovery of TtV as a new criterion extends existing knowledge and demonstrates how startups adapt prioritization to their high-uncertainty, resource-limited conditions.

RQ2.2 – How frequently are these requirements prioritization criteria discussed in the academic literature?

The frequency analysis of the 82 identified criteria revealed significant dispersion, with no single criterion dominating the literature. Cost-related factors accounted for about one fifth of all mentions, followed by value, stakeholder influence, and customer satisfaction. Temporal and performance-related criteria appeared far less often. This uneven distribution shows that while academic attention has been given to technical and managerial dimensions, temporal and outcome-oriented factors—like TtV or realized business impact—have received limited recognition.

This imbalance suggests that the literature tends to emphasize decision *processes* (how to prioritize) more than decision *consequences* (what prioritization achieves). For startups, this gap is meaningful: it shows why traditional prioritization frameworks often fail to capture the urgency and financial dependencies that characterize early-stage product decisions.

RQ3 – How can these criteria be categorized or grouped according to their underlying decision rationale?

Grouping the criteria into conceptual categories—financial, strategic, customer, technical, and temporal—revealed distinct decision rationales across literature and practice. In academic research, strategic and stakeholder-related rationales dominate, reflecting the influence of organizational alignment and customer-centric development. In contrast, the case studies showed that startups rely most on financial and temporal rationales, emphasizing what drives

immediate business validation.

This difference underscores that prioritization in startups functions less as a long-term optimization exercise and more as a short-term survival mechanism. Startups collapse cost, expected benefit, and time into a single evaluative lens focused on “what creates value fastest.” This synthesis contributes to theory by clarifying that startup prioritization can be understood as a temporal-financial heuristic rather than a purely analytical process.

RQ4 – How do financially oriented requirements prioritization criteria relate to indicators of startup performance and survival?

The integration of interview data and performance indicators demonstrated a consistent association between financial discipline and startup success. Startups that explicitly incorporated expected cost, revenue potential, and TtV into their decision-making occupied higher performance tiers. Conversely, ventures that neglected these dimensions often depleted resources early or failed to achieve market traction.

The findings indicate that financial and temporal criteria are not just descriptive of startup behavior—they are predictive of performance outcomes. Startups that prioritize features with clear, measurable financial impact tend to sustain runway longer and reach validation milestones faster. This result bridges theory and practice: while the literature has treated financial evaluation as peripheral, empirical evidence suggests it is central to survival.

Overall, the findings highlight a clear research opportunity: to formalize lightweight, finance-informed prioritization models that maintain startup agility while enabling evidence-based decision-making. Approaches that integrate financial heuristics (e.g., expected ROI, cost-to-value ratio, TtV) with agile and lean practices may provide the next methodological step for aligning requirements engineering research with startup realities.

6.6. Completeness

Within the confines of the formulated research questions (see Section 3.1), we have conducted a rigorous review exercise on requirements prioritization criteria research, with a special interest for the software startup context, resulting in the identification and selection of 40 studies that adequately address at least one of the defined research questions. Spanning from the earliest publication in 1983 to the most recent ones in 2019, the selected studies provide valuable insights into this domain. However, it is essential to acknowledge that a systematic literature review, by its nature, cannot ever be fully completed. The paradigm shift and dynamic nature of requirements prioritization, especially in the context of startup research, may have led to some relevant studies eluding our inclusion. Additionally, the decision to focus solely on English-published articles, and non-grey literature may have inadvertently omitted important and pertinent studies available in other languages. Despite these limitations, the review provides a comprehensive and valuable snapshot of the current state of requirement prioritization research, serving as a foundation for further exploration and analysis in this ever-evolving field.

6.7. Threats to Validity

This review protocol encountered various challenges that could potentially threaten its validity, including bias in publications and data extraction inaccuracies. To mitigate these challenges, a meticulous search strategy was devised, encompassing diverse literature databases, explicit selection criteria, and rigorous quality criteria. However, it is acknowledged that not all relevant research might have been captured through search terms, leading to the possibility of important studies being omitted. Furthermore, strict adherence to selection criteria (see paragraph “Define inclusion and exclusion criteria” in Section 3.2.1) and comprehensive quality assessment (see paragraph “Define quality assessment criteria” in Section 3.2.1) was enforced to minimize incorrect omissions. In the context of this structured literature review, several types of validity

were considered and addressed:

Descriptive Validity: The precision with which the research accurately portrays the collected information was ensured through the utilization of a data extraction form (see paragraph “Define data extraction monitoring” in Section 3.2.1), documenting the information essential for addressing the research questions and maintaining descriptive validity.

Theoretical Validity: To capture the intended scope of investigation effectively, efforts were made to avoid biases in study identification, maintain neutrality in inclusion and exclusion criteria, and ensure accurate data extraction and classification. Ongoing refinement of the review protocol, even using a pilot, all aspects of the protocol got checked in order to maximize the study’s theoretical validity and objectivity.

Generalizability: Given the extensive coverage of a broad research area over an extended period, the findings of this mapping study are probably generalizable, even without the inclusion of unindexed studies from grey or non-English publications.

Interpretive Validity: The involvement of researchers with expertise in software engineering and empirical research enhanced the interpretive validity of the study, minimizing the impact of personal bias on data interpretation (see Section 3.2). Here also artificial intelligence (see Section 5.1.3) got used to increase the internal consistency, in order to further reduce any personal bias on the data interpretation.

Repeatability: Documentation of the mapping procedure and outcomes based on the studied research papers contributes to the study’s repeatability, enabling replication by other researchers under similar conditions.

Although efforts have been made to ensure the validity of this study, it is recognized that other researchers might arrive at a similar list of decisions and publications but not necessarily identical to ours [3]. The inherent complexity and evolving nature of academic research necessitate a continuous and critical approach to address potential threats to validity, enhancing the reliability and robustness of future studies in this domain.

6.8. Conclusions

This study investigated how requirements prioritization criteria (RPC) are conceptualized in academic research versus applied in software startup practice, emphasizing financial and temporal decision-making under resource constraints. The systematic literature review of 40 studies revealed limited attention to the startup context, with most frameworks designed for stable, process-driven environments. Critical startup realities—such as short runways, rapid validation needs, and sensitivity to capital costs—remain largely unaddressed.

To close this gap, 34 interviews were conducted with founders from 19 software startups. Their insights reveal criteria that are both underrepresented in literature and decisive in practice. A clear and measurable business case consistently distinguished high-performing ventures, as one founder (S9) noted: “If a feature doesn’t clearly tie into a business goal, it’s a distraction we can’t afford.” Equally significant was Time to Value (TtV)—defined as the period between functional conception and the realization of business benefit. Although acknowledged in practice, TtV is absent from existing academic literature.

The results show that startups failing to consider expected cost, business case strength, and TtV systematically underperformed or went bankrupt. Academic work similarly overlooks well-established financial lenses—such as ROI, NPV, and cash flow analysis—that founders implicitly use when balancing limited capital against time pressure.

In short, the software startup perspective remains a blind spot in requirements prioritization research. Without integrating financial discipline and temporal awareness, current frameworks risk staying theoretically robust but practically irrelevant. Bridging this gap is not only an academic opportunity—it is essential for designing prioritization methods that genuinely support software startup survival and scalability.

6.9. Research Agenda

Building on these findings, future research should focus on advancing requirements prioritization (RP) methodologies that integrate Time to Value (TtV) and other financial heuristics into decision frameworks tailored to startup realities. We propose three key research directions:

RQ1. How can existing requirements prioritization methods be adapted to explicitly incorporate financial and temporal criteria—such as ROI, TtV, and cash flow analysis—to better fit high-uncertainty startup environments?

RQ2. Can lightweight, evidence-based models be developed to help startups quantify and balance cost, value, and time dimensions without compromising agility?

RQ3. To what extent do prioritization criteria predict survival across varying macroeconomic conditions, such as periods of high interest rates or constrained investment capital?

Answering these questions requires stronger academia–industry collaboration to validate and refine financially informed prioritization methods in real startup ecosystems. Such work could reshape the requirements engineering discipline by anchoring it in capital efficiency, measurable value creation, and temporal responsiveness—the key drivers of startup success in modern software economies.

Conflicts of Interest

The authors declare no conflicts of interest

Data Availability Statement

The data supporting the findings of this study are available through standard licensed open access databanks for academic research. The underlying master data and analysis are available from the authors upon reasonable request.

7. VECTR: A Lightweight Requirements Prioritization Method for Software Startups Grounded in ROI, Time-to-Value, and Confidence

Abstract⁷

Startups face critical challenges in prioritizing requirements under severe resource constraints. Although methods such as RICE and ICE are known, prior studies show that decisions often default to ad hoc judgments and stakeholder opinions. This paper introduces and validates VECTR, a lightweight prioritization framework that integrates three criteria highly relevant to startups: return on investment (ROI), Time-to-Value (TtV), and confidence. Rather than presenting these criteria as conceptually new, VECTR's contribution lies in their explicit operationalization for startup contexts and their integration into a single lightweight visualization. Seventeen practitioners—including founders, product managers, and technical leads—participated in semi-structured interviews. They first described their current prioritization practices and then evaluated VECTR through explanatory material and a proof-of-concept visualization. Results show that ROI, TtV, and confidence are already considered in practice, but inconsistently. Practitioners found VECTR intuitive, visually clear, and useful for structuring discussions, while noting limitations related to estimation effort and subjective assumptions. Most reported that it could support better data-driven decisions, though concerns remained around input estimation and assumptions made by biased humans. This study contributes a financially grounded, software-startup-suitable prioritization method and offers practitioners a practical way to allocate scarce resources toward requirements with the greatest impact, while acknowledging that the validation demonstrates perceived usefulness rather than measurable performance gains.

Keywords: requirements prioritization; software startups; product management; return on investment (ROI); Time-to-Value (TtV); confidence-based decision-making

7.1. Introduction

Startups face critical challenges in deciding which requirements to prioritize under severe resource constraints [24, 27]. These challenges include, a high pace, premature scaling, cash flow issues, difficulties in obtaining financing [50, 98], and funding shortages [51, 100]. These contribute to a 63% failing [51] rate among software startups, of which a quarter occurs in their first year [100]. In light of the aforementioned challenges, even a single failed project in early stages of the venture has the potential to push the startup into financial insolvency [28].

While frameworks such as RICE and ICE exist [212], prior research shows that decisions often default to ad hoc judgments, stakeholder influence, or gut feeling. At the same time, empirical studies have shown that financial realism, speed of impact, and evidence quality are central to survival in early-stage ventures [56]. These three dimensions—value, speed, and certainty—are commonly acknowledged in practice but are rarely operationalized together in a startup-specific way. The existing lightweight methods such as RICE and ICE rely on abstract criteria (e.g., reach, impact, and ease) that are difficult to estimate reliably in early-stage settings. Unlike generic methods such as RICE or ICE, VECTR operationalizes ROI, TtV, and confidence specifically for software-startup decision scenarios, grounding each criterion in financial and validation evidence rather than abstract scoring. By replacing abstract factors with measurable, empirically grounded ones, VECTR adapts the logic of these lightweight frameworks to the realities of resource-constrained startups. This clarification avoids overstating the method's novelty and positions its contribution appropriately.

This paper introduces and validates VECTR, a prioritization framework that integrates three criteria highly relevant to startups: return on investment (ROI) [44], Time-to-Value (TtV) [45], and confidence. VECTR aims to make trade-offs explicit through a simple visualization,

⁷ Authors: Frédéric Pattyn, Xiaofeng Wang, Geert Poels, Yannick Dillen en Peter Goetz

supporting founders and product managers in allocating scarce resources more effectively. Instead of claiming novelty at a conceptual level, the contribution lies in explicitly operationalizing these criteria for software-startup decision contexts and demonstrating that practitioners perceive this integration as intuitive and useful.

We build on previous empirical work that identified ROI, TtV, and confidence as decisive prioritization criteria, and extend it by presenting a proof-of-concept method validated with practitioners. Our research is guided by two questions:

RQ1: How do practitioners currently prioritize requirements in software startups?

RQ2: How is VECTR perceived in practice, and what are its strengths and limitations?

This study draws on feedback from 17 practitioners working in active early-stage software startups across SaaS, healthcare, fintech, biotech, and commercial software. All participating companies were in startup or early scale-up mode, and none were operating at mature or large-revenue levels. The purpose of the sample is not statistical generalization; rather, it aligns with qualitative and design-science recommendations for early-stage artifact evaluation, where the goal is to capture diverse practitioner perspectives on prioritization in resource-constrained environments. As such, the study focuses on how practitioners evaluate and interpret the VECTR method, not on the financial performance, market share, or long-term success of the participating startups.

All 17 startups were active at the time of data collection. However, long-term market outcomes fall outside the scope of this research and are not meaningful validity criteria for demonstration-based evaluation. The goal of this phase is, therefore, to assess perceived usefulness, clarity, and contextual fit—an established and appropriate objective in early early-phase artifact evaluation.

While this study focuses on prioritization within software-startup contexts, it is important to note that not all early-stage ventures require software development in their initial stages. The relevance of VECTR, therefore, arises when a startup does choose to invest in software and where resource constraints, cash-flow pressure, and runway limitations make disciplined prioritization essential. In such environments, survival depends not only on identifying the right features but also on allocating limited funding toward initiatives capable of delivering measurable value within short time horizons. By grounding prioritization in ROI, Time-to-Value, and confidence, VECTR aligns these decisions with the financial and temporal realities that strongly influence whether early-stage ventures can progress toward product-market fit and long-term viability.

The purpose of VECTR is to provide a startup-appropriate alternative to the existing lightweight methods such as RICE or ICE. While these frameworks are widely known, their criteria (e.g., “reach,” “impact,” and “ease”) are often too abstract for early-stage ventures where financial realism, delivery speed, and evidence quality are critical. Prior studies show that founders and product teams use ROI, Time-to-Value, and confidence implicitly but inconsistently. VECTR replaces the abstract components of RICE/ICE with these three operational criteria, making prioritization more explicit, structured, and aligned with the survival dynamics of software startups. Its contribution, therefore, lies in unifying well-known criteria into a startup-specific operational model that better reflects practical decision-making constraints.

The paper proceeds as follows. Section 2 reviews related work on requirements prioritization in startups and existing frameworks. Section 3 outlines our methodology. Section 4 introduces VECTR as a proof of concept. Section 5 presents findings from 17 validation semi-structured interviews, followed by discussion (Section 6), threats to validity (Section 7) and conclusions (Section 8).

7.2. Related Work

Requirements prioritization has long been recognized as a central challenge in software product management [145], with methods ranging from analytical approaches (e.g., AHP [209] and Binary Search Tree [236]) to lighter heuristics such as bubble sort, dot voting, and \$100 allocation (cumulative voting) [189, 237]. These methods provide structure, yet several studies report that, in practice, prioritization is often driven by intuition, stakeholder pressure, or short-term crises rather than systematic evaluation. This gap between available methods and actual practice is especially visible in environments with high uncertainty or frequent shifts in priorities.

In the startup context, the challenge is even more acute. Startups operate with limited resources [161, 162], extreme uncertainty [22], and short runways [56, 238], which makes prioritization not only a technical problem but also a matter of survival [28]. To maximize their operational runway, startups often prioritize RP criteria that directly reflect financial discipline, such as budget constraints [161, 168], cost–benefit ratio [11, 26], efficiency [31], anticipated maintenance cost [11] and cost–importance ratio [169]. These criteria capture “value” and “cost,” yet they remain fragmented across different studies and rarely integrate speed of impact or evidence quality explicitly.

Furthermore, prior research has emphasized the role of financial realism (e.g., ROI and cash flow) [35, 44, 173], speed of impact (Time-to-Market and Time-to-Value) [238], and confidence [10, 43] in assumptions as decisive factors for early-stage product decisions. Signed contracts or letters of intent [161] can serve as critical early signs of market traction, offering founders a higher level of confidence that can guide resource allocation. Taken together, these studies highlight the importance of value, speed, and certainty, yet no existing method explicitly operationalizes these dimensions into a lightweight, startup-suitable framework. This shift towards a financial-centric approach may yield benefits in terms of customer satisfaction [175].

While these factors are often looked at separately, no method has yet brought them together in this context. VECTR addresses this by combining ROI, TtV, and confidence into a single requirements prioritization method, offering a clear and practical way to allocate scarce resources and helping bridge the gap between academic insights and software startup practice. Having outlined the relevant literature and identified the gap, we now turn to our methodology, which details the design and steps taken to develop and assess the VECTR requirements prioritization method. Our intention is, therefore, not to claim conceptual novelty of these dimensions but to provide an integrated, operational, and empirically validated representation tailored specifically to the decision-making realities of software startups.

The existing lightweight approaches such as RICE (reach, impact, confidence, effort) and ICE (impact, confidence, ease) have proven useful for product teams, but their criteria remain abstract and difficult to quantify in early-stage ventures. To address these limitations, recent work has proposed replacing qualitative factors like “reach” and “impact” with financially and temporally measurable counterparts such as ROI and Time-to-Value, while retaining confidence. VECTR builds on this direction by explicitly defining these constructs, grounding them in startup-specific empirical evidence, and integrating them into a single visual decision-support method. This shift grounds prioritization in financial realism and temporal dynamics, making trade-offs explicit and better aligned with startup survival factors.

7.3. Research Methodology

This study employs an artifact-based research approach in which the primary aim is to develop and preliminarily evaluate a practical method — VECTR — through demonstration-based validation with practitioners [239]. Building on prior empirical work identifying ROI, Confidence, and Time-to-Value as critical startup prioritisation criteria, the method integrates

these factors into a unified visual decision-support framework. The evaluation goal was to assess perceived usefulness, clarity, and contextual fit, rather than to measure objective performance improvements — an expected and appropriate boundary for early-phase method validation. By building on prior empirical work and identifying confidence [43], ROI [44], and Time-to-Value (TtV) [45] as critical decision criteria, the proposed method integrates these factors into a unified decision-support framework named VECTR. The goal of this phase was to assess perceived usefulness, clarity, and contextual fit, rather than to measure objective performance improvements—an expected boundary of demonstration-based validation.

Demonstration-based validation was appropriate for this stage of early-phase method validation" because VECTR is an emerging artifact that has not yet been deployed longitudinally. According to the DSR guidelines, early-stage evaluation focuses on perceived usefulness, clarity, and contextual fit—especially when long-term outcome measurement is not yet feasible. The method was therefore evaluated through practitioner walkthroughs rather than extended real-world use.

VECTR helps practitioners allocate limited development resources through evidence-driven trade-offs between expected return, delivery speed, and certainty. Its practical value was evaluated through semi-structured interviews with 17 startup and product-management professionals.

Seventeen expert practitioners with experience in software startups and product management participated in the validation. Participants were recruited through purposive and snowball sampling: initial contacts were identified via professional networks (LinkedIn, ISPMA, and prior research collaborations) and subsequently expanded through participant referrals to ensure diversity.

Inclusion criteria required at least 2 years of experience in product management or founding roles within software-startup environments. One participant without such formal experience (Case P) was deliberately included because of their operational exposure to prioritization processes within a startup team. In qualitative research, such boundary cases contribute contrasting perspectives that enhance interpretive robustness; this rationale is now made explicit to avoid methodological ambiguity. We aimed for heterogeneity in domain (SaaS, healthcare, fintech, and biotech) and company stage within early-stage and early scale-up boundaries to capture varied perspectives on prioritization practices.

The final group included founders, product managers, and technical leads across multiple domains such as SaaS, software, and healthcare. To maintain confidentiality, participants are referred to anonymously (A–Q); see Table 46.

Table 44: Overview of studied cases, interviewees

Case	Domain	Role	Startup Experience
A	Healthcare	Product manager	Active in startup context
B	Commercial	Lead developer	Active in startup context
C	Healthcare	Founder and CEO	Active in startup context
D	Food	CEO and product manager	Active in startup context
E	Software	Founder and CEO	Active in startup context
F	Software	Lead developer	Active in startup context
G	Software	Founder and CEO	Active in startup context
H	Biotech	Founder and CEO	Active in startup, 2nd startup venture

I	Software	Co-founder	Serial entrepreneur and investor in startups
J	Healthcare	Co-founder	Active in startup context and venture scout
K	Fintech	Founder	Serial entrepreneur
L	Software	Founder	Active in startup context
M	Software	Chief product officer	Serial PM growing startups to scale-ups
N	Software	Chief product officer	Active in startup context
O	Software	Chief information officer	Ex-founder, growing startups to scale-ups
P	Software	Product owner	No experience
Q	Software	Chief information officer	Serial entrepreneur and board member at multiple startups

Purposive and snowball sampling were chosen to capture the diversity of roles and contexts rather than representativeness, consistent with qualitative research standards for exploratory artifact evaluation.

Although Table 46 may appear at first glance as a basic overview of respondents, its purpose in this study is to contextualize the design-science evaluation rather than to function as a questionnaire or survey instrument. The table summarizes the characteristics of the practitioners whose qualitative insights form the empirical basis for both research questions. In the first stage of each interview, participants described their current prioritization practices, which informed RQ1. In the second stage, the same practitioners evaluated VECTR after being introduced to its logic and visualization, which informed RQ2. Table 46 therefore links directly to the research design by providing a structured overview of the diverse startup contexts, roles, and experiential backgrounds through which VECTR was assessed. This context is essential, as the evaluation of early-stage design-science artifacts depends on practitioners' interpretive insights rather than large-scale quantitative data, and Table 46 clarifies the practical environments against which VECTR was judged.

Each semi-structured interview lasted 45–60 min and was conducted via a Teams video call. Once consent was given, all interviews were recorded, transcribed, and coded thematically. Practitioners were first asked to describe their current prioritization practices and challenges. VECTR was then introduced through explanatory material and proof-of-concept visualization. Participants were guided through how ROI, TtV, and confidence are assessed and combined before being invited to evaluate the method's relevance, strengths, limitations, and adoption potential in their context.

The evaluation followed a demonstration-based validation, which focused on perceived usefulness rather than long-term performance effects. Because participants did not apply VECTR to their own backlogs, the findings represent an immediate interpretation of the method rather than the measured impact. This limitation is inherent to early-phase artifact evaluation and aligns with established standards for early-phase artifact evaluation.

All interviews were recorded, transcribed, and coded thematically. Coding was structured around the two research questions: RQ1 (current prioritization practices) and RQ2 (evaluation of VECTR). Themes were developed iteratively, and the results are presented in Section 5.

In terms of data analysis, all interviews were transcribed verbatim and coded using an inductive thematic approach. Two researchers independently coded three transcripts to calibrate the codebook, resolving differences through discussion. The remaining transcripts were coded by one researcher and reviewed by the other to ensure consistency. Themes were iteratively refined around the research questions (RQ1 and RQ2) until no new codes emerged, indicating

thematic stability rather than statistical generalizability, which is appropriate for exploratory qualitative validation.

7.4. Software Startup Requirements Prioritization Method: VECTR

VECTR is a software-startup context-appropriate requirements prioritization method designed for high-uncertainty, resource-constrained environments.

To make the underlying logic explicit, VECTR evaluates each requirement along three estimation dimensions: (1) Time-to-Value, which reflects how quickly a feature can deliver meaningful benefits once the development begins; (2) confidence, which captures the strength and quality of the evidence supporting the requirement; and (3) return on investment, which represents the expected value relative to cost or effort. By combining these three perspectives into a single visualization, VECTR enables teams to compare alternatives transparently, surface trade-offs, and prioritize work that offers fast impact, strong validation, and disproportionate value. This framing supports clear, evidence-informed decision-making while keeping the method lightweight and startup-appropriate.

Its contribution lies not in introducing new criteria, but in explicitly operationalizing ROI, TtV, and confidence into a lightweight visualization aligned with how startup teams naturally think about value, speed, and certainty.

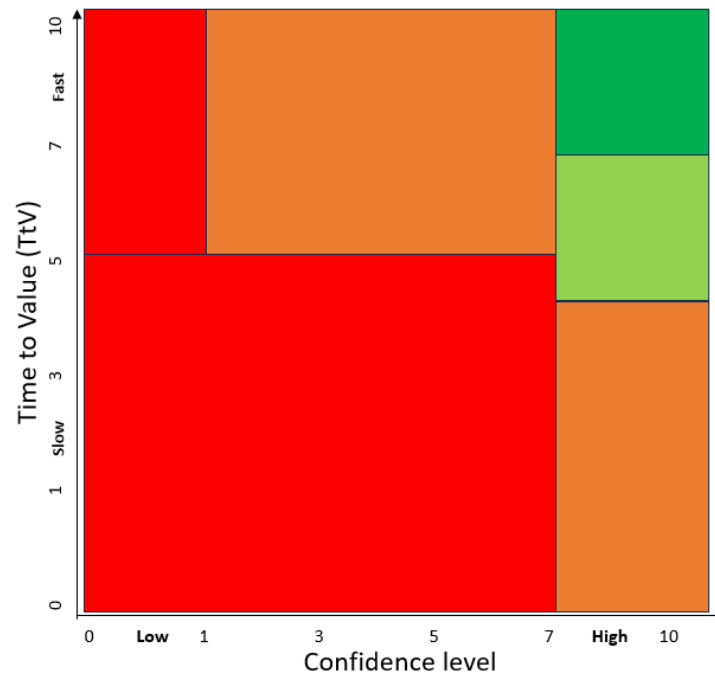
The method is operationalized as a two-dimensional prioritization map. The x-axis represents confidence in the underlying assumptions of a requirement, assessed through evidence quality and validation progress; scoring ranges from intuition (low) to revenue evidence (high).

To support consistent confidence assessment, teams should reference specific evidence artifacts at each level. Intuition-level confidence reflects founder belief or team assumptions with no external validation (artifacts: internal brainstorm notes and strategy documents). User-interview confidence requires documented customer conversations confirming the problem or need (artifacts: interview transcripts, customer quotes, and survey responses showing demand). Prototype evidence confidence requires demonstration that users engage with a tangible solution (artifacts: clickthrough rates on mockups, waitlist signups, pilot user feedback, and usability test results). Revenue-evidence confidence requires demonstrated willingness to pay or actual purchasing behavior (artifacts: pre-orders, paid pilots, recurring subscriptions, and measured revenue lift). For example, a feature might begin at the intuition level when a founder hypothesizes that “small businesses need better invoice tracking.” After conducting eight customer interviews revealing consistent pain points, confidence moves to the user interview level. Building a clickable prototype and observing 40% of test users completing the invoice flow elevates it to prototype evidence. Finally, launching a paid pilot with five customers generating \$2000 MRR establishes revenue-evidence confidence. This progression framework helps teams track validation maturity over time and identify where additional evidence gathering is most needed.

The y-axis represents Time-to-Value (TtV). $\text{Time-to-Value (TtV)} = \text{Time-to-Market} + \text{Time-to-Business-Value}$, where business value may denote the time until measurable outcomes such as first revenue, user-retention improvement, or KPI achievement (e.g., 10% revenue increase). To reduce ambiguity and support consistent estimation, business value should be defined relative to the requirement’s primary objective: for revenue-generating features, it reflects time to first paying customer or measurable revenue increase (e.g., 10% lift); for retention features, it reflects time to observable churn reduction or engagement improvement (e.g., 15% increase in DAU); for cost-reduction features, it reflects time to measurable operational savings (e.g., 20% support ticket reduction); for enablement features, it reflects time to adoption by internal teams or users (e.g., 50% team adoption). Teams should anchor estimates to concrete metrics observable within their typical measurement cycles (weekly for B2C and monthly/quarterly for B2B) rather than abstract notions of “value delivered.” This operational definition ensures TtV

remains interpretable across different requirement types while acknowledging that exact measurement remains context-dependent (see Figure 19).

Figure 21: Conceptual view of the VECTR framework. Color zones indicate strategic priority: green = high confidence + fast TtV (highest priority); red = low confidence + slow TtV (lowest priority).



In practice, ROI (1) is estimated over a defined time horizon H (typically 3–12 months for startups) as $\text{ROI (\%)} = (\text{Expected Profit over } H - \text{Investment Cost}) / \text{Investment Cost} \times 100$. Profit may include increased revenue, churn reduction, or cost savings, while Investment Cost covers both development and operational expenses (OPEX). ROI may also incorporate opportunity cost if relevant for portfolio trade-offs. Negative ROI values indicate non-viable initiatives.

$$\text{ROI (\%)} = \frac{\text{Expected Profit over horizon } H}{\text{Total Investment Cost}} \times 100$$

(1)

ROI is visualized as the diameter of the bubble, reflecting the ratio of expected gains to investment costs (1), with 100% ROI serving as the base size (D^0) (2) (see Figure 20). This is how to interpret the visualization: consider two hypothetical requirements positioned on the map. Requirement A (“mobile app onboarding redesign”) has ROI = 50%, TtV = 2 months, and confidence = “prototype evidence.” It appears as a smaller bubble (diameter $0.7\times$ the baseline) in the lower-left quadrant, reflecting moderate speed but lower financial return and moderate validation. Requirement B (“enterprise API integration”) has ROI = 200%, TtV = 6 months, and confidence = “revenue evidence.” It appears as a larger bubble (diameter $1.4\times$ the baseline) in the upper-right quadrant, reflecting higher financial impact but slower delivery and stronger validation. Comparing these visually, teams can quickly see that requirement A offers faster time-to-impact but lower returns with less certainty, while requirement B requires patience but delivers stronger validated returns. This encoding allows decision-makers to

assess trade-offs at a glance while understanding that bubble size corresponds linearly to ROI magnitude relative to the 100% baseline.

Figure 22: Visual encoding in VECTR: confidence on the x-axis, Time-to-Value on the y-axis, and ROI represented by bubble size

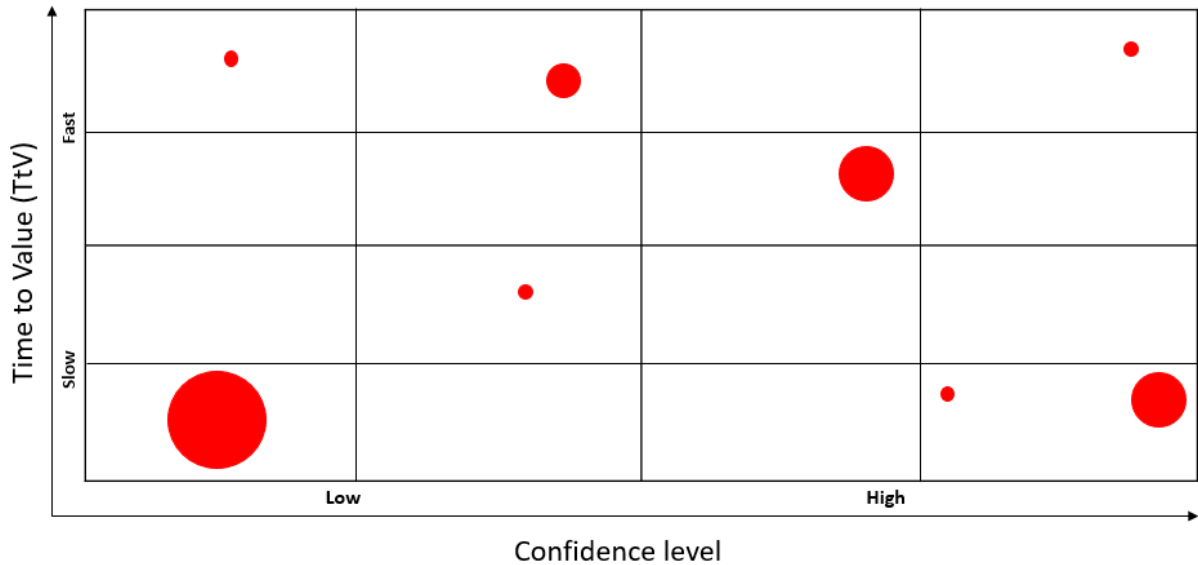
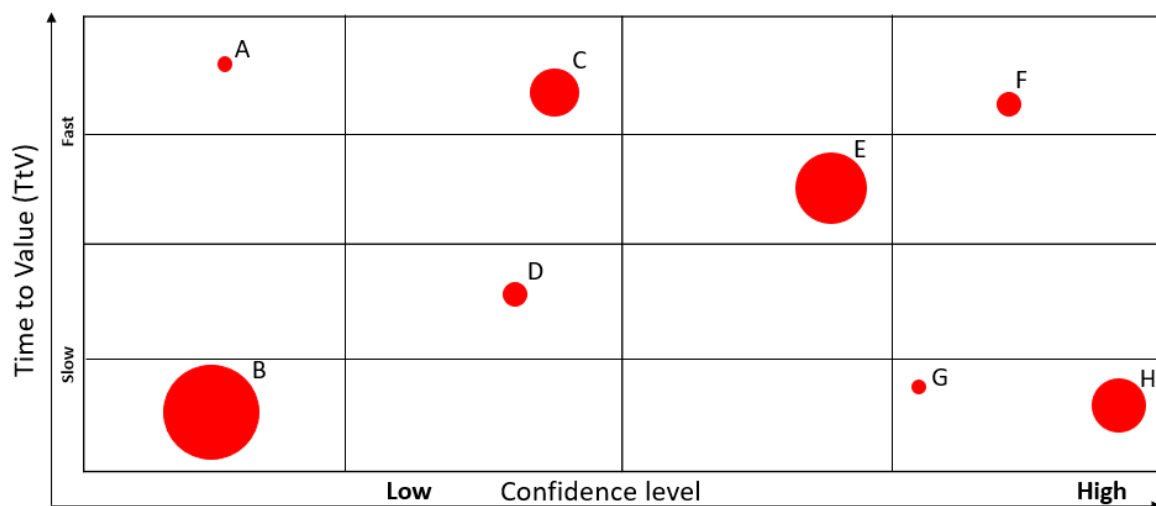


Figure 23: Example VECTR map. Larger bubbles in the upper-right (high ROI + high confidence + fast TtV) represent highest-priority requirements.



Variable	A	B	C	D	E	F	G	H
Time to Value (TtV)	9	2	8	4	7	7	2	2
Confidence level	0.03	0.03	1	0.5	3	7	7	10
Return (% ROI)	50%	500%	250%	100%	200%	150%	75%	250%
Score	0.135	0.3	20	2	50	73.5	10.5	50

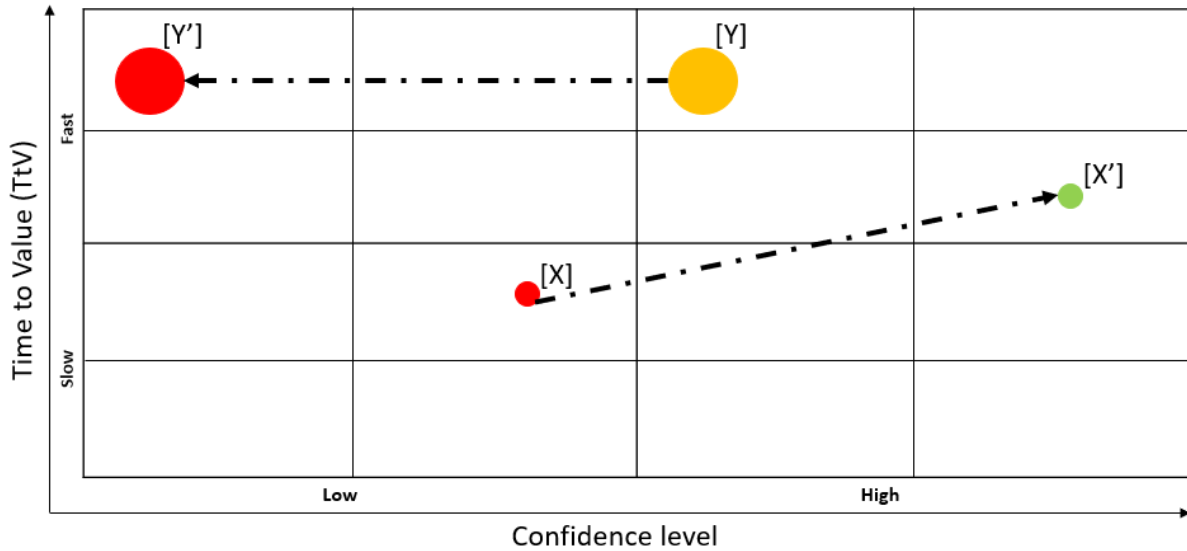
$$\text{Diameter} = D^0 x \sqrt{\frac{ROI (\%)}{100}} \quad (2)$$

To preserve perceptual accuracy, marker area is scaled in proportion to ROI% rather than diameter alone (see Figure 20). Requirements with $ROI \leq 0$ can be displayed as thin rings or crossed markers to signal non-viability, while extremely high ROI values are capped to maintain readability. Furthermore, a small reference legend or example bubble sizes can help prevent misinterpretation of perceptual scaling, and the model uses area-based scaling to avoid visually exaggerating differences between items.

It is important to vet the incoming ideas (see Figure 21) and, when conviction is strong, invest in refining their confidence levels.

As illustrated in Figure 22, a requirement may shift from a low-confidence “red” state (**position X**) to a validated, high-confidence “green” position (**position X'**). Conversely, ideas that appear highly promising with fast Time-to-Value (**position Y**) may, after validation, reveal critical weaknesses and move from an “orange” to “red” position (**position Y'**), sparing the startup from costly mistakes.

Figure 24: Dynamics of confidence within VECTR. Example scenario: Requirement [X] gains confidence through validation and moves from red to green ($[X \rightarrow X']$); requirement [Y] loses confidence and shifts from orange to red ($[Y \rightarrow Y']$).

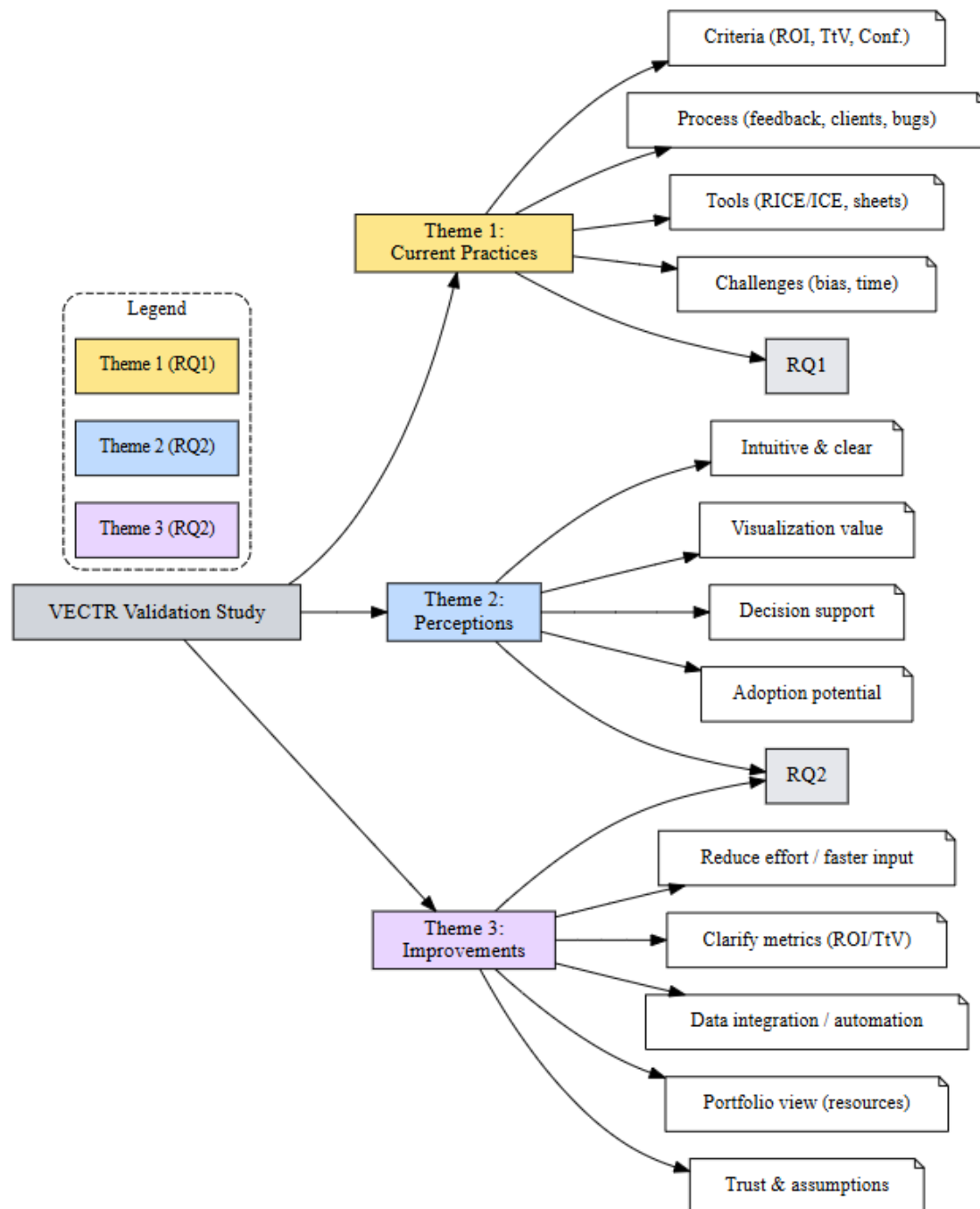


In this way, VECTR emphasizes that validation is not only about ranking requirements but also about preventing misallocation of scarce resources.

7.5. Findings

The findings provide insight into how practitioners perceive and evaluate the VECTR framework when applied to the challenges of software startup prioritization. Rather than focusing on individual experiences, the analysis highlights recurring themes and contrasting viewpoints (Figure 22) that shed light on both the strengths and limitations of VECTR in practice. To maintain clarity, the results are presented according to the two research questions, with overarching patterns summarized in Figure 23 and further illustrated through selected excerpts.

Figure 25: Thematic structure of findings, linking current practices (RQ1) and practitioner evaluation of VECTR (RQ2).



7.5.1. RQ1—Current Prioritization Practices

Participants described pragmatic, multi-layered decision processes. Financial factors such as cash, ROI, revenue, and funding availability were central: “It’s a game of survivability, so revenue solves a lot of problems” (Case I). Time-related factors, including Time-to-Market and Time-to-Value, also shaped decisions, alongside ability to execute with available resources. Confidence was often inferred indirectly, for example through the recurrence of customer feedback: “If the same feedback comes from multiple data points, it becomes a theme... that rises to the top” (Case A). Urgent production issues frequently overrode plans: “If there are bugs in production, I leave everything and fix them within 30 min” (Case D).

Frameworks such as RICE or ICE were occasionally used but rarely sustained. Final decisions often reverted to stakeholder judgment or politics: “ICE gets theoretically used, but the core management team basically does the prioritization based on their opinions” (Case O). Several participants acknowledged the resulting bias, noting that “human biases are the trickiest part in all of this” (Case H).

Overall, ROI, Time-to-Value, and confidence were already part of decision-making, but were applied inconsistently and in ad hoc ways. Table 45 summarizes the subthemes of current practices with representative quotes. These findings confirm that, while practitioners implicitly consider the three VECTR dimensions, they lack a unified or systematic way of bringing them together—reinforcing the need for a lightweight operational method.

Table 45: Themes of current prioritization practices (RQ1) with representative quotes

Subtheme	Quotes
Criteria (Confidence)	I would say that in the early days, talk to as many people as you possibly can about your idea. It will save you in the long run (Case K). We really want to look at the confidence level of the projects (Case Q).
Criteria (TtV)	Generate traction because of the runway limitations (Case M). Then you want to look at the ability to execute. How hard is it to execute (Case I)? Our biggest criteria obviously was to look at the speed of innovation. (Case J). It’s about revenue, right? How many customers are signing up (Case K).
Criteria (ROI)	What’s the ROI in the projects (Case Q)? Probably a good rule of thumb is to look what is the idea with the highest margin (Case I)?
Tools	Word (Case K)—Excel (Case A, H, I, J, L, M, N, O, P and Q)—Notion (Case B)—GitHub (Case C, D and E)—Trello (Case G)—PowerPoint (Case M, N, O, P and Q). RICE gets applied here (Case N). ICE gets theoretically used, but the core management team basically does the prioritization based on their opinions (Case O).
Process (Feedback)	So, if we get feedback about something from one person, OK, that’s a data point. But if that seems same, feedback comes in and it’s coming in from multiple, you know data points, it becomes a theme. And so we start to put more weight on reoccurring things (Case A). You want to, when starting out, the smallest modicum of value that you can get out and test the market with and see if you can get initial traction (Case I). we went to [potential] customers, we went to people and we spoke to them. We started seeing a pattern (Case J). We always consider feedback calls, even if you have paying customers (Case K).
Process (Clients)	Finding a balance between paying customers to support your runway, and be true to your product market fit (Case O).

Challenge (Cashflow)	Runway management; Absolutely. You need to survive long enough (Case H). The last thing is how much money do we have? That's the biggest thing. (Case J). The hardest problem is you got to balance out your cash needs to get to scale. That's the name of the game. If you can't manage a P&L, good luck running a business (Case K). Finding product market fit which has a strong impact on your cashflow as a startup (Case N).
Challenge (Revenue)	If you're already in the market and revenue generating, well then the challenge is you know, keeping that engine going and growing (Case I).
Challenge (People)	It's a matter of bringing the right people, not having too many people, but having the right number of people (Case I). We don't invest enough time and energy in the validation phase, to increase our confidence level enough to counteract the stakeholders opinions (Case M). How to handle the opinions of some very strong stakeholders (Case Q)?
Challenge Time-to-Market (TtM)	Another challenge that we are facing is the pace of delivery (Case M).
Challenge (Bias)	Bias selection of prio since you wear multiple hats (Case F).

7.5.2. RQ2—Evaluation of VECTR

Practitioners responded positively to VECTR. All 17 participants described the framework as intuitive: “Very simple logic” (Case H), “hits the mark for sure” (Case K), and “a perfect fit for our projects” (Case L). Several noted its alignment with lean startup thinking and its potential in roadmap discussions.

In terms of decision support, ten participants believed VECTR would improve prioritization, five were undecided, and two disagreed. Supporters highlighted its ability to depersonalize debates: “Having something to point to helps people check their own egos... more data-driven versus emotion-driven” (Case H). Skeptics stressed the subjectivity of inputs and the risk of manipulating numbers to justify preferences. Six participants were not asked about adoption due to time constraints near the interview end; these sessions focused on conceptual feedback instead. This distribution reflects perceived usefulness rather than proven performance impact, which aligns with the demonstration-based validation design.

Participants also valued the visualization. Plotting confidence (x-axis), TtV (y-axis), and ROI as bubble size was seen as effective in surfacing quick wins and clarifying trade-offs: “You can just focus on the big dots at the top—these are the things you should build next” (Case G). Some saw it as particularly valuable for quarterly planning and portfolio reviews.

Regarding adoption potential, five said they would adopt VECTR directly, four were considering, two declined, and six were not asked. Adoption was seen as most likely in strategic contexts such as board meetings and planning sessions, while barriers included estimation effort and the need for lightweight tooling. Several participants also noted that VECTR could be more vulnerable to subjective estimation or political scoring in larger or more mature organizations, reinforcing its suitability primarily for early-stage, resource-constrained environments.

Participants also suggested improvements. They called for faster input mechanisms (e.g., table uploads), clearer metric definitions (especially ROI, including churn reduction, operational expenditures (OPEX), and opportunity cost), integration with existing datasets for automation, and portfolio-level capacity overlays. Others highlighted the importance of transparency to build trust and reduce bias.

Overall, VECTR was perceived as intuitive, useful, and visually compelling, with clear decision-support value. Challenges remain around effort, assumptions, and integration. Tables 48 summarize the main themes of VECTR's evaluation with representative quotes. Taken together, the findings suggest that VECTR works best when decisions revolve around balancing runway, speed of learning, and evidence strength, and it may be less effective in contexts with rigid governance, heavy bureaucracy, or entrenched stakeholder politics.

Table 46: Practitioner evaluation of VECTR (RQ2): strengths and illustrative quotes

Subtheme	Quotes
Intuitive and Clear	The output for me is perfect, easy, clear (Case E). Very straightforward & super hands on (Case F). That visualization is very, very helpful (Case G). Yeah, I mean, it's very simple (Case H).
Visualization Value	Very systematic way of showing info (Case A). Nice graph (Case D).
Decision Support	It just removes every subjective view... and tells you what you what you should do next (Case G). Good way of structuring, and ... prioritize ... and determine what to build next first (Case I). This thing which will give you a better idea of which products to do work first on (Case J). It makes decision-making easy (Case L). Having something to point to, I guess would help people check their own egos (Case H). Data-driven vs. emotion driven (Case H).
Adoption Potential	... this would be extremely valuable for founders; get an idea of how to start (Case K)
Reduce Effort/Faster input	Input data via GitHub or directly in a table, not using GPT, takes to much time (Case D) Just work a little bit more on the question side (Case E) Takes too long maybe... (Case F).
Clarify Metrics	Helping users understand what those metrics are (Case C) You'd probably have to have a ... explainer video along with it for those companies. This would be interesting for educational purposes, because not all startup founders know or understand this already (Case I).
Data Integrations/ Automation	Link it to the enormous ocean of [research & healthcare] information (Case H). Give me a roadmap based on what my financial resources (Case K). I'm interested in the next steps as well; how much resources are required within the portfolio considering the opportunities (Case N)?

Portfolio View	More room for like for example, it was there maybe different options for people to look into different kind of diagram (Case C).
Trust and Assumptions	Take it with a grain of salt, because you're going to have to make some assumptions (Case I). Human biases are the trickiest part (Case H).

Beyond perceived usefulness, it is important to consider when VECTR is most appropriate and when it may fail. The method appears especially suited for early-stage contexts where forcing assumptions into the open and aligning cross-functional teams on priority rationale are critical. By making confidence explicit, VECTR encourages teams to articulate the evidence (or lack thereof) behind each requirement, which supports faster learning cycles and reduces attachment to pet projects. However, the method may be vulnerable to several failure modes. In more mature or politically charged organizations, scoring could become subject to gaming or strategic inflation, where stakeholders artificially boost ROI or confidence estimates to secure resources for preferred initiatives. Similarly, VECTR risks false precision if teams treat rough estimates as hard numbers, losing sight of the inherent uncertainty in early-stage forecasting. The visualization's simplicity—while useful for communication—may also obscure important nuances such as sequencing dependencies, technical risks, or strategic imperatives that defy quantification.

Finally, VECTR assumes teams have sufficient domain knowledge and customer proximity to make meaningful estimates; in contexts where customer access is limited or market understanding is shallow, the inputs may reflect wishful thinking rather than evidence-grounded judgment. These boundary conditions suggest that VECTR is best deployed as a discussion catalyst and alignment tool rather than a deterministic ranking algorithm.

7.5.3. Illustrative Cases Demonstrating Real-World Success Using ROI, TtV, and Confidence

Several cases from the study show that decision-making practices grounded in ROI, Time-to-Value, and validation confidence already contribute to meaningful progress in real-world startup environments. Although VECTR was not applied directly within these companies, practitioners' own descriptions reveal that these three criteria frequently underpinned successful product and business outcomes.

In Case A, the team systematically converted customer feedback into confidence signals by treating individual comments as isolated data points and elevating recurring patterns to strategic priorities. This disciplined approach enabled them to identify the most relevant features and integrate them into the roadmap, resulting in stable product direction and features that consistently generated value for customers.

In Case E, early market research and customer interviews informed the initial feature set. After release, the team continuously monitored duplicate requests and weighted them according to frequency, allowing them to prioritize enhancements that were most likely to strengthen product-market fit. This evidence-driven process supported both a successful first version and sustained traction as new requirements emerged.

Case F illustrated a structured OKR-driven approach in which new initiatives were evaluated based on their expected ROI and their contribution to quarterly objectives. The combination of financial reasoning, feasibility assessments, and input from individual contributors helped ensure that development capacity was focused on initiatives with the highest strategic value, contributing to predictable execution and internal alignment.

A similar emphasis on financially grounded prioritization appeared in Case H, where long-term survival depended on managing cash burn and extending runway. The founder described

routinely running simulations to determine which projects offered the highest expected return or survivability benefit. This disciplined selection process helped the organization navigate a difficult investment climate and make progress despite sustained operational losses.

In Case I, the team prioritized work that protected and expanded the existing revenue streams. Maintaining execution capacity in sales, marketing, and product development required a focus on features that could deliver value quickly. This Time-to-Value mindset enabled the company to continue growing even as external funding conditions became increasingly uncertain.

Case K demonstrated a highly systematic use of validation as a confidence-building mechanism. Before committing to development, the team sought extensive user input—ranging from dozens to hundreds of potential users—combined with prototypes, waiting lists, and pre-orders to establish evidence of demand. This approach supported early revenue generation, reduced wasteful development, and helped maintain strong founder alignment, which was seen as essential to sustained progress.

In Case M, runway limitations drove the need to prioritize features capable of generating traction quickly. Strategic alignment and stakeholder agreement were considered crucial, and initiatives were selected based on how rapidly they could produce measurable customer or market impact. This emphasis on rapid Time-to-Value supported early adoption and reduced the risk of delayed results.

Finally, Case N highlighted the importance of selecting initiatives that contributed to product-market fit and stronger unit economics. Decisions were framed around strategic ROI, ensuring that development effort was concentrated on features that meaningfully improved revenue efficiency or long-term viability.

Across these examples, founders and product leaders consistently linked their most meaningful progress—such as improved traction, clearer roadmaps, early revenue generation, and financial stability—to practices that align closely with ROI, Time-to-Value, and confidence. These patterns provided additional empirical grounding for the relevance of the three criteria that VECTR brings together into a unified decision-support method.

7.6. Discussion

This study highlights how startups already rely on ROI, Time-to-Value, and confidence when prioritizing, but in fragmented and inconsistent ways. Frameworks such as RICE or ICE are known but applied inconsistently, and decisions frequently revert to opinion or short-term pressure. By integrating the three most salient criteria into a single visualization, VECTR directly addresses this gap, offering a lightweight, transparent approach tailored to startup realities. Rather than introducing fundamentally new criteria, VECTR's contribution lies in operationalizing these dimensions in a startup-specific manner that practitioners find intuitive and aligned with their decision-making logic.

Compared to existing methods, VECTR places stronger emphasis on financial realism (ROI), speed of impact (TtV), and evidence quality (confidence). Practitioners validated these dimensions as central to their daily decisions but appreciated how VECTR unifies them in a form that reduces subjectivity and anchors discussion. Its visualization highlights quick wins while also exposing longer-term or riskier options, thereby supporting both immediate survival and strategic planning. This supports the claim that lightweight decision support can help make assumptions explicit—an important need surfaced repeatedly in prior studies on startup uncertainty. It also aligns with recent findings that startups increasingly rely on AI-generated summaries, analysis prompts, and structured decision aids to reduce ambiguity and maintain consistent reasoning under conditions of uncertainty.

Compared with RICE/ICE, VECTR offers stronger alignment with startup decision contexts by explicitly linking prioritization to financial and temporal survival metrics. Participants recognized that ROI and TtV introduce a “runway logic” absent in other frameworks. However,

several also noted that, like any scoring-based model, VECTR can be influenced by subjective estimation or biased assumptions, especially in organizations where strong stakeholders dominate prioritization debates. This boundary condition limits the tool's applicability in more mature, politically complex environments.

The role of AI in supporting prioritization: prior research shows that AI-based tools can help founders structure early decisions, accelerate evidence gathering, and surface assumptions that might otherwise remain implicit. AI can assist with estimation consistency, synthesizing inputs from customer feedback, and producing first-pass reasoning or alternative interpretations. Practitioners in earlier studies also reported that AI-generated visualizations and structured outputs help reduce cognitive load and improve communication between product teams and stakeholders. These capabilities complement VECTR's goal of making prioritization more transparent and scalable, and future work could explore deeper integration between VECTR and AI-supported estimation or evidence generation systems.

Practical implications: for startup founders and product managers, VECTR offers a defensible way to explain and justify prioritization choices in strategic contexts such as board meetings, investor updates, and quarterly planning. Its visual simplicity provides a common language that helps reduce political debate, while clarifying the trade-offs between speed, risk, and return. By supporting data-driven rather than intuition-driven decision-making, VECTR strengthens alignment between teams and stakeholders. At the same time, VECTR should be used as a facilitation aid rather than a prescriptive-ranking mechanism; the risk of "false precision" arises if teams treat rough estimates as exact values.

At the same time, the findings underline limitations and improvement needs. Practitioners pointed to the effort required for input estimation, the subjectivity of assumptions, and the need for clearer metric definitions—particularly regarding ROI, which should capture not only revenue but also churn reduction, OPEX, and opportunity costs. They also saw value in linking VECTR to external datasets to automate inputs and in extending the framework with portfolio-level resource constraints. These suggestions connect naturally to potential AI-based enhancements, such as AI-assisted estimation, automated evidence aggregation, and integration with real-time usage or customer data. Additionally, novelty bias and immediacy bias may have influenced participant enthusiasm, as the artifact was shown immediately before feedback collection.

Finally, the validation highlights that decision-support tools are not a substitute for managerial judgment, but a complement. VECTR should, therefore, be viewed as a way to structure conversations and reduce bias rather than a prescriptive formula. This distinction is critical for ensuring both credibility and adoption. Future work—ideally involving longitudinal use of VECTR in real startup backlogs—will be able to assess whether it improves alignment, decision speed, and confidence development over time. A subsequent research direction is also to investigate whether AI-supported versions of VECTR can further enhance estimation quality, reduce uncertainty, and support evidence-based prioritization in dynamic startup environments.

7.7. Threats to Validity

Internal validity: social desirability bias may have led participants to overstate the method's usefulness. Novelty bias and immediacy bias may also have influenced enthusiasm, as the artifact was introduced directly before the evaluation.

Construct validity: as interviews were demonstration-based, perceived usefulness may not equate to effective decision support. The study captures practitioners' interpretations of the method rather than the measured improvements in prioritization accuracy or outcomes. Because VECTR was not deployed in a real startup workflow, no claims about performance

improvement can yet be made. Evaluating objective performance effects requires longitudinal or in situ application, which falls outside the scope of early-stage design-science evaluation. External validity: the purposive sample of 17 experts' limits generalizability beyond similar early-stage ventures.

Reliability: while thematic saturation was achieved, coding involved some subjective interpretation. Saturation reflects stability of themes within this sample, not exhaustiveness across all startup contexts.

7.8. Conclusions

This study examined how early-stage software startups make prioritization decisions and proposed VECTR as a lightweight method grounded in ROI, Time-to-Value, and confidence. The interview findings confirm that practitioners already rely on these three dimensions—often implicitly—to balance limited resources, short runways, and high uncertainty. VECTR builds on these practices by offering a simple structure and visualization that make evidence strength and expected value explicit.

Practitioners described VECTR as intuitive and helpful in clarifying assumptions, aligning teams, and depersonalizing roadmap discussions. The method appears especially suitable for early-stage environments where speed of learning, clear trade-offs, and transparent decision-making are critical. At the same time, the study highlights important limitations: VECTR depends on the quality of the underlying estimates, may be vulnerable to subjective scoring, and is less effective in mature organizations with rigid governance or complex dependencies. Overall, the findings suggest that VECTR provides a practical, startup-appropriate way to structure prioritization conversations, especially when teams face constraints on time, budget, and evidence. Rather than serving as a deterministic scoring model, VECTR is best used as an alignment and discussion tool that helps founders expose assumptions, focus on fast and validated value, and avoid overcommitting to poorly supported initiatives.

Author Contributions: Conceptualization, Frédéric Pattyn; Methodology, Frédéric Pattyn; Formal analysis, Frédéric Pattyn; Investigation, Frédéric Pattyn; Writing – original draft, Frédéric Pattyn; Writing – review & editing, Xiaofeng Wang, Geert Poels, Yannick Dillen and Peter Goetz; Visualization, Frédéric Pattyn; Project administration, Frédéric Pattyn. All authors have read and agreed to the published version of the manuscript.

Funding:. This research received no external funding.

Data Availability Statement:. The data presented in this study are available from the corresponding author upon reasonable request. Due to company-sensitive and confidential information contained in the dataset, the data cannot be made publicly available.

Conflicts of Interest:. Author Peter Goetz is employed by the company New Data Research Institute. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

8. Conclusion

8.1. Research Results and Scientific Contributions

In the introduction, this dissertation set out to address the overarching problem that software startups lack systematic, evidence-based knowledge and practical tools to guide effective product management activities and prioritization decisions under conditions of high uncertainty and resource scarcity. To investigate this, four research questions were formulated in Chapter 1 and investigated across Chapters 2 to 7 using a multi-method research approach combining systematic literature reviews, surveys, expert interviews, multiple case studies, and design-oriented method development — organized across three phases: (1) Exploration, (2) Empirical Deep Dive, and (3) Synthesis.

RQ1. Which product management activities are most critical in software startups?

The findings demonstrate that product management in software startups is substantially broader, more fragmented, and more generalist in nature than commonly reflected in existing product management literature. Chapter 2 identified 107 distinct product management activities across six domains through a systematic literature review, resulting in the Pragmatic Framework for Product Managers (PFPM). The findings revealed that product management extends far beyond traditional roadmap ownership or requirements management and includes executive leadership, validation, sales, marketing, stakeholder management, customer support, and operational coordination.

RQ2: How do product management activities in software startups differ from those in mature organizations?

Chapter 3 further demonstrated that product managers in both startup and mature contexts predominantly operate as generalists rather than specialists. Using the Herfindahl–Hirschman Index (HHI) as a quantitative measure of role concentration, the findings showed that 94% of product managers exhibited low concentration scores, indicating broad role dispersion across multiple activity domains. Of the five specialists identified in the sample, four were founders operating in startup environments — all four concentrating their attention in the domain of Product Strategy and Planning. This domain also attracted the largest share of attention across the full sample, accounting for approximately 30% of total activity allocation regardless of organizational context. Together, these findings suggest that the critical differentiator between startup and mature product management lies not in which activities are performed, but in the asymmetric consequences of incorrect decisions within Product Strategy and Planning.

RQ3. Which criteria are most relevant for prioritizing product decisions in software startups?

The findings from Chapter 4 indicate that stronger-performing startups were not necessarily characterized by conducting more validation activities, but rather by obtaining stronger forms of evidence earlier in the startup lifecycle. The multiple case study revealed that all investigated startups engaged in some form of validation behaviour. However, higher-performing startups more frequently secured direct financial commitment from external stakeholders before committing substantial resources to MVP development. In particular, Level 6 evidence — obtaining paying customers prior to full MVP development — emerged as the strongest differentiator between higher- and lower-performing startups.

The findings also demonstrated that lower forms of validation evidence, such as surveys, interviews, waitlists, or expressions of interest, often created a false sense of certainty when not followed by stronger market commitment signals. This led to the development of the

Hierarchy of Validation, a bidirectional process model that structures validation activities according to the strength of the evidence generated. The results therefore suggest that startup validation should not primarily be evaluated based on the quantity of experiments performed, but rather on the commitment level of the evidence obtained.

The findings from Chapters 5 and 6 demonstrate that requirements prioritization in software startups differs significantly from the assumptions commonly reflected in existing literature. Chapter 5 provided breadth through a large-scale international practitioner survey, while Chapter 6 provided depth through triangulation between systematic literature review findings and longitudinal founder interviews.

Chapter 5 showed that startup practitioners place substantially greater emphasis on financial survival, revenue generation, and short-term impact than practitioners in mature organizations. Chapter 6 further demonstrated that Time-to-Value (TtV) emerged as a highly important but significantly underrepresented prioritization criterion within existing scholarly literature. The findings additionally showed that many existing prioritization methods implicitly assume organizational stability, predictable environments, and sufficient resources — assumptions that frequently do not hold in startup environments characterized by constrained runway and evolving requirements. Together, the findings indicate that startup prioritization behaviour is fundamentally shaped by survival-oriented decision-making rather than optimization-oriented planning.

RQ4. How can evidence-based insights into activities and criteria be translated into actionable frameworks and tools for startups?

The findings from Chapter 7 resulted in the development of VECTR, a lightweight requirements prioritization method specifically designed for software startup environments. VECTR combines three core prioritization dimensions: Return on Investment (ROI), Time-to-Value (TtV), and Confidence. Unlike many existing prioritization methods that primarily focus on value estimation or stakeholder weighting, VECTR explicitly incorporates uncertainty and temporal urgency into prioritization decisions. The practitioner evaluations demonstrated that participants perceived the framework as intuitive, practical, and aligned with real-world startup constraints. The visual and dynamic nature of the framework allowed startups to reassess priorities continuously as validation confidence evolved over time.

Taken together, the dissertation demonstrates that software startup product management is fundamentally shaped by uncertainty, resource scarcity, validation pressure, and survival-oriented prioritization behaviour. The findings further show that many existing product management and requirements prioritization theories remain primarily grounded in assumptions associated with mature organizational environments.

This dissertation contributes to the state-of-the-art in software startup product management and requirements prioritization research through six interconnected scientific contributions. The dissertation should not be interpreted as a disconnected bundle of papers, but as a cumulative research program in which each chapter answers part of the overall problem statement from a different methodological angle. The program begins by mapping and measuring the product management role, then investigates startup validation and prioritization logic under uncertainty, and finally synthesizes these findings into a startup-appropriate prescriptive method.

A recurring finding across the recent literature included in the dissertation review materials is that current requirements prioritization and product management research remains strongly

technique-oriented, frequently focused on mature or enterprise-scale organizational environments, and still fragmented regarding startup-specific decision-making realities. Recent systematic reviews and method papers continue to emphasize prioritization algorithms, AI-supported ranking approaches, stakeholder weighting, and large sets of requirements, while relatively limited attention is given to the actual prioritization criteria used by startup practitioners operating under uncertainty and survival constraints [240-245]. This strengthens the novelty of the dissertation: its contribution is not another prioritization algorithm, but an empirically grounded account of what product managers and startup practitioners actually prioritize, why these priorities differ by context, and how these insights can be translated into a lightweight startup-appropriate method.

Contribution 1 — Chapter 2: The Pragmatic Framework for Product Managers (PFPM)

Recent research confirms that the gap this framework addresses remains open. Grigoryan, Martin [13] documented the continuing fragmentation of product management responsibilities (2025), and Estrada-Esponda, Matturro [246] confirm that startup environments require broad combinations of technical, strategic, and interpersonal skills that existing frameworks do not adequately capture (2025). Neither study provides the structured, startup-specific activity map that PFPM offers. The contribution is therefore not consolidation for its own sake, but the provision of a first evidence-based reference model that defines what product management actually looks like before product-market fit.

Contribution 2 — Chapter 3: Quantifying Product Management Role Concentration Through HHI

Recent literature confirms the absence of robust quantitative approaches for measuring product management role dispersion. Grigoryan, Martin [13] emphasize the growing complexity and overlap of product management responsibilities but do not provide a formalized quantitative measurement approach. This dissertation therefore extends the state-of-the-art by introducing a replicable and ratio-scale-compatible mechanism for measuring role concentration empirically, creating a foundation for future comparative research investigating how product management attention allocation evolves across organizational contexts and AI-supported environments.

Contribution 3 — Chapter 4: The Hierarchy of Validation

The findings demonstrate that stronger-performing startups were not characterized by conducting more validation activities, but by obtaining stronger forms of commitment-based evidence earlier in the lifecycle. The contribution therefore lies less in identifying individual validation techniques — which were broadly known in startup literature — and more in uncovering the process logic and directional sequencing through which stronger-performing startups approached validation. The Hierarchy of Validation contributes a novel conceptual mechanism for understanding validation maturity and decision confidence that is grounded inductively in 29 semi-structured interviews with 16 startup founders.

Recent startup-oriented requirements engineering research by Estrada-Esponda, Matturro [246] continues to emphasize the importance of uncertainty management and market interaction in startup environments, yet does not provide a structured evidence hierarchy for

comparing validation strength. The Hierarchy of Validation therefore contributes a novel and empirically grounded framework for pre-build decision-making that connects directly to the confidence dimension of VECTR in Chapter 7. A central implication of the framework is that validation evidence is not merely informational but gating: at L1 and L2, the uncertainty level has not decreased sufficiently to warrant the resource commitment that substantial development requires. Startups operating solely on informal discussions, surveys, or expressions of interest have not yet obtained the level of external commitment that justifies building. The finding that none of the Tier 3 or out-of-business startups progressed beyond L2 before building suggests that the absence of stronger market evidence is not a gap to be filled after development begins, but a signal that development should not yet have started.

Contribution 4 — Chapter 5: Context-Dependent Requirements Prioritization in Software Startups

Existing requirements prioritization literature has historically focused strongly on prioritization techniques, ranking algorithms, stakeholder weighting mechanisms, and optimization approaches. None of the prior large-scale studies had empirically compared prioritization criterion weighting across organizational maturity levels, and dominant frameworks — including ICE, RICE, and MoSCoW — were not validated for startup contexts. Chapter 5 addresses this gap by providing one of the most comprehensive empirical datasets on requirements prioritization to date, drawing on 171 practitioners across 37 countries and 51 industries, and demonstrating statistically significant differences in how financial, temporal, and strategic criteria are weighted across startup and mature company contexts.

The 2025–2026 literature strongly confirms that this gap persists. Ais, De Souza [240] cover 45 new prioritization studies and 32 techniques from 2021–2025 while providing limited insight into contextual prioritization behaviour across organizational types. Tasneem, Zulzalil [242] review 76 agile prioritization studies and acknowledge that effectiveness depends on organizational context, yet do not empirically investigate startup-specific criteria. Limaylla-Lunarejo, Condori-Fernandez [241] confirm that AI-based prioritization research treats criteria mainly as algorithmic inputs rather than empirical objects of study. Albidah, Abbadeni [243] explicitly focus on medium-to-large mobile software organizations, confirming the persistent absence of startup-context prioritization research in the empirical literature. Wang, Khalajzadeh [247] demonstrate significant heterogeneity in prioritization preferences across user subgroups within a single domain (mHealth), reinforcing that context-specificity in requirements prioritization extends beyond organizational maturity and remains empirically relevant. These parallel findings validate the central contribution of Chapter 5: shifting the focus from 'which prioritization technique performs best' toward 'which prioritization criteria matter most under startup conditions.'

Contribution 5 — Chapter 6: Time-to-Value as a Missing Prioritization Criterion

The findings from Chapters 5 and 6 demonstrate that startup practitioners frequently prioritize the speed at which value can be realized, validated, or monetized. However, existing scholarly literature largely underrepresents temporal urgency as an explicit prioritization dimension. The 82-criterion literature review conducted in Chapter 6 confirmed that Time-to-Value (TtV) — defined as the period between the release of a feature and the

moment its intended business value is largely realized — was entirely absent from the scholarly record.

Recent state-of-the-art reviews strongly reinforce this gap. Ais, De Souza [240] do not identify TtV as a dominant standalone criterion despite covering 45 recent studies. Limaylla-Lunarejo, Condori-Fernandez [241] focus predominantly on stakeholder weighting, business value estimation, and algorithmic optimization rather than temporal startup survival dynamics. Raj and Deora [248] demonstrate that recent AI-driven prioritization models remain heavily focused on feature optimization and ML-based ranking rather than integrated financial-temporal decision-making.

The contribution is not merely the naming of a new criterion, but the identification of a broader causal mechanism linking faster value realization to startup survival. The findings suggest that shorter TtV contributes financially through runway preservation, informationally through faster learning and uncertainty reduction, and indirectly through increased investor confidence and execution signaling. In environments increasingly influenced by AI-supported software development where Time-to-Market may continue to decrease substantially, TtV may become the more strategically relevant differentiator — a point reinforced by the growing speed of AI-assisted development pipelines documented in recent technique papers [244, 245].

Contribution 6 — Chapter 7: VECTR — A Startup-Oriented Requirements Prioritization Method

VECTR extends the state-of-the-art by integrating ROI, TtV, and Confidence into a single lightweight framework specifically designed for uncertainty-intensive startup environments. The contribution is cumulative: Confidence is operationalized through the validation logic of Chapter 4, ROI is reframed through the survival-oriented founder logic of Chapters 5 and 6, and TtV emerges as the newly identified criterion absent from prior literature. VECTR therefore represents not a reweighting of existing methods such as ICE or RICE, but a startup-native decision logic grounded in the empirical findings of the broader research programme.

Together, the six contributions of this dissertation demonstrate that software startup product management requires theories, frameworks, and prioritization methods explicitly adapted to environments characterized by uncertainty, resource scarcity, rapid iteration, and survival-oriented decision-making. The dissertation therefore contributes both descriptive and prescriptive advances that extend current product management and requirements prioritization research beyond predominantly mature organizational assumptions.

8.2. Practical contributions

Beyond its scientific contributions, this dissertation also provides several practical contributions for founders, product managers, accelerators, consultants, and software startup ecosystems.

PFPM offers founders and early-stage product managers a structured overview of the breadth of product management responsibilities within startup environments. By making these responsibilities explicit, the framework helps startups identify overlooked activities and improve resource allocation.

The HHI-based role concentration analysis provides organizations with a mechanism for evaluating whether product management attention is overly fragmented or overly specialized, supporting organizational design decisions and future AI-assisted workload allocation.

The Hierarchy of Validation provides startups with a structured mechanism for evaluating the strength of market evidence before committing substantial development resources, helping founders avoid false validation signals and improve investment discipline.

The findings on startup prioritization behaviour provide evidence-based insight into how startups make survival-oriented decisions under uncertainty, supporting investors, incubators, and advisors in evaluating startup maturity and prioritization quality.

VECTR provides a lightweight and practitioner-oriented prioritization method that can be applied directly within software startup environments. Because the framework integrates value, speed, and confidence, it supports startups in balancing commercial viability with uncertainty reduction and learning acceleration.

8.3. Research limitations

Sample scope. Several empirical studies relied on purposive sampling and relatively limited sample sizes. While appropriate for exploratory and context-sensitive investigation, broader replication studies would strengthen generalizability.

Contextual heterogeneity. Startup environments are highly heterogeneous. Differences in funding stage, industry, founder experience, geography, and market maturity may influence prioritization behaviour and validation practices. Although the dissertation focused on software startups as a shared context, variation within this category remains substantial.

Self-reported data. Several findings rely partially on self-reported practitioner perceptions and retrospective interpretations. Although triangulation and mixed-method approaches were used to mitigate this limitation, self-reporting bias cannot be fully excluded.

Evolving context. The startup ecosystem evolves rapidly, particularly under the increasing influence of AI-supported software development and decision-making. Some product management activities and prioritization practices may continue to shift over time in ways that extend beyond the scope of this dissertation.

VECTR validation scope. VECTR was intentionally designed as a lightweight practical framework rather than a mathematically optimized algorithm. While this improves usability and startup applicability, future studies may further investigate formal optimization or automation extensions.

8.4. Future research

Broader organizational contexts. Future studies could extend the scope to scale-up environments, hardware startups, or other industries with longer development cycles, higher capital intensity, or regulatory constraints. Comparative research across industries and geographies would test the robustness of the findings.

Longitudinal perspectives. Longitudinal case studies could reveal how product management priorities shift as startups progress through their lifecycle — from TtV-driven survival

decisions to ROI-driven scaling strategies — building a richer understanding of the transitions in product management across organizational maturity stages.

Validation and longitudinal study of the Hierarchy of Validation. Future research could further operationalize the Hierarchy of Validation and investigate whether specific validation sequences correlate with long-term startup outcomes in larger and more diverse samples.

TtV as a research construct. Additional research is needed to investigate Time-to-Value as a prioritization construct, including its relationship with runway management, investor behaviour, and AI-accelerated software development — particularly given the growing speed of AI-assisted development pipelines documented in recent literature [244].

VECTR longitudinal evaluation. VECTR could be evaluated through longitudinal field studies within active startup environments and compared empirically against alternative prioritization approaches to assess its impact on survival, runway extension, and product-market fit.

AI-native development environments. The increasing rise of AI-native software development creates opportunities to investigate whether prioritization, validation, and product management decision-making become partially automated or augmented through intelligent systems, and how frameworks such as VECTR and the Hierarchy of Validation hold up in this context.

8.5. Closing reflection

Software startups operate in environments where time, money, and certainty are continuously constrained. Under these conditions, survival often depends less on building more features and more on making better decisions regarding what to build, when to build it, and how strongly those decisions are supported by evidence.

This dissertation argued that many existing product management and requirements prioritization theories remain strongly influenced by assumptions associated with mature organizations. Software startups, however, operate under fundamentally different conditions characterized by uncertainty, rapid iteration, incomplete information, and survival pressure.

By combining systematic literature investigation, empirical research, quantitative analysis, validation theory, and practical method design, this dissertation contributes toward a more startup-appropriate understanding of product management and prioritization.

Ultimately, the dissertation demonstrates that startup product management is not merely about feature delivery or roadmap planning. It is fundamentally about navigating uncertainty through structured prioritization, validation, and evidence-based decision-making. The central argument emerging across the research programme is therefore that startups do not primarily fail because they are unable to build products, but because they build the wrong things under conditions where the cost of being wrong is existential rather than incremental.

References

1. Ebert, C., *The impacts of software product management*. 2006.
2. Tyagi, R.K. and M.S. Sawhney, *High-performance product management: the impact of structure, process, competencies, and role definition*. Journal of Product Innovation Management, 2010. **27**(1): p. 83-96.
3. Barney, S., A. Aurum, and C. Wohlin, *A product management challenge: Creating software product value through requirements selection*. Journal of Systems Architecture, 2008. **54**(6): p. 576-593.
4. Maglyas, A., U. Nikula, and K. Smolander, *What are the roles of software product managers? An empirical investigation*. Journal of Systems and Software, 2013. **86**(12): p. 3071-3090.
5. Krishnan, V. and K.T. Ulrich, *Product development decisions: A review of the literature*. Management science, 2001. **47**(1): p. 1-21.
6. Cooper, R.G. and E.J. Kleinschmidt, *Benchmarking the firm's critical success factors in new product development*. Journal of Product Innovation Management: An International Publication of the Product Development & Management Association, 1995. **12**(5): p. 374-391.
7. Springer, O. and J. Miler. *The role of a software product manager in various business environments*. in *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*. 2018. IEEE.
8. González Vázquez, J.M., J. Sauer, and H.-J. Appelrath, *Methods to manage information sources for software product managers in the energy market*. Business & Information Systems Engineering, 2012. **4**(1): p. 3-14.
9. Gorchels, L., *Transitioning from engineering to product management*. Engineering Management Journal, 2003. **15**(4): p. 40-47.
10. Cooper, R.G., *From experience: the invisible success factors in product innovation*. Journal of product innovation management, 1999. **16**(2): p. 115-133.
11. Wohlin, C. and A. Aurum. *Criteria for selecting software requirements to create product value: An industrial empirical study*. in *Value-based software engineering*. 2006. Springer.
12. Springer, O., J. Miler, and M. Wróbel, *Strategies for dealing with software product management challenges*. IEEE Access, 2023. **11**: p. 55797-55813.
13. Grigoryan, K., et al., *Product management: Tasks, roles, and the importance of data*. Procedia CIRP, 2025. **136**: p. 838-843.
14. Calantone, R.J., C.A. Di Benedetto, and R. Divine, *Organisational, technical and marketing antecedents for successful new product development*. R&D Management, 1993. **23**(4): p. 337-351.
15. Ernst, H., *Success factors of new product development: a review of the empirical literature*. International journal of management reviews, 2002. **4**(1): p. 1-40.
16. Ahmad, M.O., et al., *An empirical study of portfolio management and Kanban in agile and lean software companies*. Journal of Software: Evolution and Process, 2017. **29**(6): p. e1834.
17. Rodriguez, P., C. Urquhart, and E. Mendes, *A Theory of Value for Value-based Feature Selection in Software Engineering*. IEEE Transactions on Software Engineering, 2020.
18. Bekkers, W., et al. *A framework for process improvement in software product management*. in *Systems, Software and Services Process Improvement: 17th European Conference, EuroSPI 2010, Grenoble, France, September 1-3, 2010. Proceedings 17*. 2010. Springer.
19. Lehtola, L. and M. Kauppinen, *Suitability of requirements prioritization methods for market-driven software product development*. Software Process: Improvement and Practice, 2006. **11**(1): p. 7-19.
20. Astuti, T., A.F. Helmi, and B. Riyono, *Differences in Organizational Behavior amongst Startup and Established Company: A Literature Review*. Buletin Psikologi, 2023. **31**(2).
21. Svahnberg, M., et al., *A systematic review on strategic release planning models*. Information and software technology, 2010. **52**(3): p. 237-248.
22. Sommer, S.C., C.H. Loch, and J. Dong, *Managing complexity and unforeseeable uncertainty in startup companies: An empirical study*. Organization Science, 2009. **20**(1): p. 118-133.
23. Maital, S. and E. Barzani, *Why startups fail: a survey of empirical studies*. 2021, Samuel Neaman Institute. <https://www.neaman.org.il/en/why-startups-fail-a-...>
24. Giardino, C., et al. *Key challenges in early-stage software startups*. in *International conference on agile software development*. 2015. Springer.
25. Gralha, C., et al. *The evolution of requirements practices in software startups*. in *Proceedings of the 40th International Conference on Software Engineering*. 2018.
26. Melegati, J., et al., *A model of requirements engineering in software startups*. Information and software technology, 2019. **109**: p. 92-107.
27. Paternoster, N., et al., *Software development in startup companies: A systematic mapping study*. Information and Software Technology, 2014. **56**(10): p. 1200-1218.

28. Giardino, C., X. Wang, and P. Abrahamsson. *Why early-stage software startups fail: a behavioral framework*. in *Software Business. Towards Continuous Value Delivery: 5th International Conference, ICSOB 2014, Paphos, Cyprus, June 16-18, 2014. Proceedings* 5. 2014. Springer.
29. Tokarev, B. *Comparative Analysis of the Product Management Application in Startups of Different Types*. in *Proceedings of the International Scientific Conference "Smart Nations: Global Trends In The Digital Economy" Volume 1*. 2022. Springer.
30. Klotins, E., M. Unterkalmsteiner, and T. Gorschek, *Software engineering in start-up companies: An analysis of 88 experience reports*. Empirical Software Engineering, 2019. **24**(1): p. 68-102.
31. Tripathi, N., et al., *An anatomy of requirements engineering in software startups using multi-vocal literature and case survey*. Journal of Systems and Software, 2018. **146**: p. 130-151.
32. Gutbrod, M., J. Münch, and M. Tichy. *How do software startups approach experimentation? Empirical results from a qualitative interview study*. in *Product-Focused Software Process Improvement: 18th International Conference, PROFES 2017, Innsbruck, Austria, November 29–December 1, 2017, Proceedings* 18. 2017. Springer.
33. Melegati, J., et al. *MVP and experimentation in software startups: a qualitative survey*. in *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. 2020. IEEE.
34. Mota, R.d.O., *A framework for time-to-market reduction in startups*. 2022.
35. Gompers, P.A., et al., *How do venture capitalists make decisions?* Journal of Financial Economics, 2020. **135**(1): p. 169-190.
36. Creswell, J.W. and V.L.P. Clark, *Designing and conducting mixed methods research*. 2017: Sage publications.
37. Morse, J.M., *Principles of mixed methods and multimethod research design*. Handbook of mixed methods in social and behavioral research, 2003. **1**: p. 189-208.
38. Yin, R.K., *Case study research and applications*. Vol. 6. 2018: Sage Thousand Oaks, CA.
39. Corbin, J.M., *Basics of qualitative research: Grounded theory procedures and techniques*. 1990: Sage.
40. Denzin, N.K. and Y.S. Lincoln, *The Sage handbook of qualitative research*. 2011: sage.
41. Pattyn, F., *The Critical Role of Product Managers and Their Responsibilities in Software Startups: A Systematic Literature Review*. American Journal of Engineering and Technology Management, 2024. **9**(4).
42. Pattyn, F., *Innovating the Future: Unveiling the Initial Iteration of the Pragmatic Framework for Product Managers*. American Journal of Engineering and Technology Management, 2024.
43. Frédéric Pattyn, U.R., *Driving startup success through hierarchy of validation*. PREPRINT 2025.
44. Pattyn, F. and U. Rafiq. *Smart Scaling for Software Startups Through Financial Requirements Prioritization Criteria*. in *15th International Conference on Software Business (ICSOB 2024)*. 2024. Utrecht: Springer.
45. Pattyn, F. and P. Goetz, *Time to Value (TtV) as a Missing Requirements Prioritization Criterion for Startup Survival: Insights from Founders vs. Systematic Literature Review*. International Journal of Sustainability Management and Information Technologies, 2025. **11**(1): p. 45-66.
46. Pattyn, F., et al., *VECTR: A Requirements Prioritization Method for Software Startups Based on What Really Matters: Time to Value, Confidence level and Return On Investment*. Computers, 2026. **15**(2).
47. Pattyn, F. *Enhancing Startup Success Rates: Towards a Pragmatic Framework for Product Managers (PFPM)*. in *IEEE 31st International Requirements Engineering Conference (RE)*. 2023. IEEE.
48. Crowne, M. *Why software product startups fail and what to do about it. Evolution of software product development in startup companies*. in *IEEE International Engineering Management Conference*. 2002. IEEE.
49. Pattyn, F. *Preliminary Structured Literature Review Results using ChatGPT: Towards a Pragmatic Framework for Product Managers at Software Startups*. in *IEEE 31st International Requirements Engineering Conference Workshops (REW)*. 2023. IEEE.
50. Michael, F. *Cash Flow: The Reason 82% of Small Businesses Fail*. 2020 [cited 2023 01/03/2022]; Available from: <https://preferredcfo.com/cash-flow-reason-small-businesses-fail/>.
51. Kyril, K., *Startup Failure Rate: How Many Startups Fail and Why in 2023?*, in *Failory*. 2022, Failory: 14/12/2022.
52. Komssi, M., et al., *Roadmapping problems in practice: value creation from the perspective of the customers*. Requirements Engineering, 2015. **20**(1): p. 45-69.
53. Hujainah, F., et al., *Software requirements prioritisation: a systematic literature review on significance, stakeholders, techniques and challenges*. IEEE Access, 2018. **6**: p. 71497-71523.
54. Azar, J., R.K. Smith, and D. Cordes, *Value-oriented requirements prioritization in a small development organization*. IEEE software, 2007. **24**(1): p. 32-37.
55. Souder, W.E., D. Buisson, and T. Garrett, *Success through customer-driven new product development: a comparison of US and New Zealand small entrepreneurial high technology firms*. Journal of Product

- Innovation Management: an international publication of the product development & management association 1997. **14**(6): p. 459-472.
56. Pattyn, F. *The Hidden Costs of Ignoring Cash Flow: A Call for Strategic Requirements Prioritization at Startups during an Era of Rising Interest Rates*. in *IEEE 31st International Requirements Engineering Conference Workshops (REW)*. 2023. IEEE.
 57. Eisenmann, T.R., *Determinants of early-stage startup performance: Survey results*. Harvard Business School Entrepreneurial Management Working Paper, 2020(21-057).
 58. Höst, M., et al., *Exploring bottlenecks in market-driven requirements management processes with discrete event simulation*. Journal of Systems and Software, 2001. **59**(3): p. 323-332.
 59. Ebert, C., *Software product management*. IEEE Software, 2014. **31**(3): p. 21-24.
 60. Kittlaus, H.-B., *Software product management and agile software development: conflicts and solutions*. Software for People: Fundamentals, Trends and Best Practices, 2012: p. 83-96.
 61. Kitchenham, B., et al., *Systematic literature reviews in software engineering—a systematic literature review*. Information and software technology, 2009. **51**(1): p. 7-15.
 62. Achimugu, P., et al., *A systematic literature review of software requirements prioritization research*. Information and software technology, 2014. **56**(6): p. 568-585.
 63. Parikh, N.A., *Managing ai-first products: Roles, skills, challenges, and strategies of ai product managers*. IEEE Engineering Management Review, 2025.
 64. Kunwar, P.J., et al., *Productization of carbon handprint—A product management perspective*. Cleaner Engineering and Technology, 2025. **25**: p. 100903.
 65. Prevel Katsanis, L., *Some effects of changes in brand management systems: issues and implications*. International Marketing Review, 1999. **16**(6): p. 518-532.
 66. Wiśniowska-Kielian, B., B. Filipek-Mazur, and F. Gambús, *Possibility of Using the By-Product of Fatty Acid Extraction from Fish in Fertilization as an Element of the Circular Economy*. Agronomy, 2024. **14**(1): p. 127.
 67. Donoghue, I., L. Hannola, and J. Papinniemi, *Product lifecycle management business transformation in an engineering technology company*, in *Product Lifecycle Management (Volume 4): The Case Studies*. 2019, Springer. p. 185-200.
 68. Gorschek, T. and C. Wohlin, *Requirements abstraction model*. Requirements Engineering, 2006. **11**(1): p. 79-101.
 69. Miler, J. and M. Menjega-Schmidt, *Evaluation of Selected UX Techniques by Product Managers—A Preliminary Survey*, in *Integrating Research and Practice in Software Engineering*. 2019, Springer. p. 159-169.
 70. González Vázquez, J.M., J. Sauer, and H.-J. Appelrath, *Methods to manage information sources for software product managers in the energy market: A reference model catalog for the energy market*. Business & Information Systems Engineering, 2012. **4**(1): p. 3-14.
 71. Chen, S. and M. Tseng, *Defining specifications for custom products: a multi-attribute negotiation approach*. CIRP annals, 2005. **54**(1): p. 159-162.
 72. Wu, H., et al., *GeoColab: an LLM-based multi-agent collaborative framework for geospatial code generation*. International Journal of Digital Earth, 2025. **18**(2): p. 2569405.
 73. Sameti, A., *Strategic horizon of market success: comparing alternative “strategic landscapes” of the new product development process*. Journal of Product & Brand Management, 2023. **32**(7): p. 1063-1081.
 74. Stevens, V.J., et al., *Design and implementation of an interactive website to support long-term maintenance of weight loss*. Journal of medical Internet research, 2008. **10**(1): p. e1.
 75. Adler, P.S., et al., *From project to process management: An empirically-based framework for analyzing product development time*. Management science, 1995. **41**(3): p. 458-484.
 76. Lehong, S., E. Dube, and G. Angelopoulos, *An investigation into the perceptions of business stakeholders on the benefits of enterprise architecture: The case of Telkom SA*. South African Journal of Business Management, 2013. **44**(2): p. 45-56.
 77. Avlonitis, G.J. and D.A. Karayanni, *The impact of internet use on business-to-business marketing: examples from American and European companies*. Industrial marketing management, 2000. **29**(5): p. 441-459.
 78. Argouslidis, P.C., G. Baltas, and A. Mavrommatis, *An empirical investigation into the determinants of decision speed in product elimination decision processes*. European management journal, 2015. **33**(4): p. 268-286.
 79. Dzamashvili Fogelström, N., et al., *The impact of agile principles on market-driven software product development*. Journal of software maintenance and evolution: Research and practice, 2010. **22**(1): p. 53-80.

80. Vish Krishnan, V., et al., *Inclusive innovation: Product innovation in technology supply chains*. Manufacturing & Service Operations Management, 2019. **21**(2): p. 327-345.
81. Trieflinger, S., et al., *The discovery effort worthiness index: How much product discovery should you do and how can this be integrated into delivery?* Information and software technology, 2023. **157**: p. 107167.
82. Chen, S., et al., *Automatically distilling storyboard with rich features for android apps*. IEEE Transactions on Software Engineering, 2022. **49**(2): p. 667-683.
83. Disch, M., *Estimating the costs of product development projects based on external data: The new product development cost benchmarking method*. IEEE Transactions on Engineering Management, 2023. **71**: p. 5117-5128.
84. Kittlaus, H.-B., *Usability from a Product Manager's Perspective I*. 46. WI-MAW-Rundbrief.
85. Barhydt, J., *Psychological Safety in Startup Organizations*. 2023: Pepperdine University.
86. Gupta, V., L. Rubalcaba, and C. Gupta, *Connecting Dots Between Entrepreneurs, Research Publishers, and Software Engineering Researchers: An Outcome of Mixed Methods Empirical Research*. IT Professional, 2023. **25**(1): p. 68-80.
87. Kleinaltenkamp, M., K. Prohl-Schwenke, and L. Elgeti, *The Rise of a New Business Function: Customer Success (Management)*, in *Customer Success Management: Helping Business Customers Achieve Their Goals*. 2023, Springer. p. 1-6.
88. Wu, Z., *Analysis of Market Demand and Skills of Product Manager*.
89. Frédéric Pattyn, P.G., *Perceived Product Management Priorities in Software Startups Perceived Product Management Priorities in Software Startups A Data-Driven Reduction of the Pragmatic Framework for Product Managers from 171 Practitioners*, in *The 9th International Workshop on Software-intensive Business (IWSiB 2026)*. 2026: Rio de Janeiro, Brazil.
90. Ocasio, W., *Towards an attention-based view of the firm*. Strategic management journal, 1997. **18**(S1): p. 187-206.
91. Roelandt, J., P. Andries, and M. Knockaert, *The contribution of board experience to opportunity development in high-tech ventures*. Small Business Economics, 2022. **58**(3): p. 1627-1645.
92. Rhoades, S.A., *The herfindahl-hirschman index*. Fed. Res. Bull., 1993. **79**: p. 188.
93. Brezina, I., et al., *Herfindahl–Hirschman index level of concentration values modification and analysis of their change*. Central European journal of operations research, 2016. **24**(1): p. 49-72.
94. Krivka, A., *On the concept of market concentration, the minimum Herfindahl-Hirschman index, and its practical application*. 2016.
95. Weinstock, D.S., *Using the Herfindahl Index to measure concentration*. The Antitrust Bulletin, 1982. **27**(2): p. 285-301.
96. Pattyn, F., *Towards a Pragmatic Framework for Product Managers at Software Startups: A Systematic Literature Review*. Preprint for journal of Information and Software Technology, 2024.
97. Ferreira, D. and R.K. Sah, *Who gets to the top? Generalists versus specialists in managerial organizations*. The RAND journal of economics, 2012. **43**(4): p. 577-601.
98. Fatema, K., M.M. Syeed, and M.S.U. Miah, *Demography of Startup Software Companies: An Empirical Investigation on the Success and Failure*. International Journal of Computer Applications, 2020. **975**: p. 8887.
99. Klotins, E., M. Unterkalmsteiner, and T. Gorschek, *Software engineering antipatterns in start-ups*. IEEE Software, 2018. **36**(2): p. 118-126.
100. Kolmar, C. *Must know startup statistics (2023): average time to reach profitability at a startup*. [Website] 2023 [cited 2023 01/03/2023]; Available from: <https://www.zippia.com/advice/start-up-profitability-statistics/>.
101. Brielmaier, C. and M. Friesl, *The attention-based view: Review and conceptual extension towards situated attention*. International Journal of Management Reviews, 2023. **25**(1): p. 99-129.
102. Kleinknecht, R., et al., *An attention-based view of short-termism: The effects of organizational structure*. European Management Journal, 2020. **38**(2): p. 244-254.
103. Rerup, C., *Attentional triangulation: Learning from unexpected rare crises*. Organization Science, 2009. **20**(5): p. 876-893.
104. Shapiro, C., *The 2010 horizontal merger guidelines: From hedgehog to fox in forty years*. Antitrust LJ, 2010. **77**: p. 49.
105. Kolasky, W.J. and A.R. Dick, *The merger guidelines and the integration of efficiencies into antitrust review of horizontal mergers*. Antitrust LJ, 2003. **71**: p. 207.
106. Carlton, D.W. and J.M. Perloff, *Modern industrial organization, global edition*. Higher education, 2015. **4**.
107. Stevens, S.S., *On the theory of scales of measurement*. Science, 1946. **103**(2684): p. 677-680.

108. Ukav, I., *Market structures and concentration measuring techniques*. Asian Journal of Agricultural Extension, Economics & Sociology, 2017. **19**(4): p. 1-16.
109. Calkins, S., *The new merger guidelines and the Herfindahl-Hirschman Index*. Calif. L. Rev., 1983. **71**: p. 402.
110. Susilo, Y.O. and K.W. Axhausen, *Repetitions in individual daily activity–travel–location patterns: a study using the Herfindahl–Hirschman Index*. Transportation, 2014. **41**(5): p. 995-1011.
111. Melegati, J. and A. Goldman. *Requirements engineering in software startups: A grounded theory approach*. in *2016 International Conference on Engineering, Technology and Innovation/IEEE International Technology Management Conference (ICE/ITMC)*. 2016. IEEE.
112. Hokkanen, L., K. Kuusinen, and K. Väänänen. *Minimum viable user experience: A framework for supporting product design in startups*. in *Agile Processes, in Software Engineering, and Extreme Programming: 17th International Conference, XP 2016, Edinburgh, UK, May 24-27, 2016, Proceedings 17*. 2016. Springer.
113. Maturro, G., et al. *Minimum Viable Product Creation and Validation in Software Startups*. in *2021 XLVII Latin American Computing Conference (CLEI)*. 2021. IEEE.
114. Alonso, S., et al., *A systematic mapping study and practitioner insights on the use of software engineering practices to develop MVPs*. Information and Software Technology, 2023. **156**: p. 107144.
115. Ros, R., E. Bjarnason, and P. Runeson, *A theory of factors affecting continuous experimentation (FACE)*. Empirical Software Engineering, 2024. **29**(1): p. 21.
116. Tripathi, N., et al., *Startup ecosystem effect on minimum viable product development in software startups*. Information and Software Technology, 2019. **114**: p. 77-91.
117. Fagerholm, F., et al., *The RIGHT model for continuous experimentation*. Journal of Systems and Software, 2017. **123**: p. 292-305.
118. Ries, E., *How today's entrepreneurs use continuous innovation to create radically successful businesses*. The lean startup, 2011.
119. Auer, F. and M. Felderer. *Current state of research on continuous experimentation: a systematic mapping study*. in *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. 2018. IEEE.
120. Melegati, J., X. Wang, and P. Abrahamsson. *Hypotheses engineering: first essential steps of experiment-driven software development*. in *2019 IEEE/ACM Joint 4th International Workshop on Rapid Continuous Software Engineering and 1st International Workshop on Data-Driven Decisions, Experimentation and Evolution (RCoSE/DDrEE)*. 2019. IEEE.
121. McKinley, D. *Design for Continuous Experimentation: Talk and Slides*. 2012 [cited 2024 12/09/2024]; Available from: <https://mcfunley.com/design-for-continuous-experimentation>.
122. Insights, C., *The Top 12 Reasons Startups Fail*. 2021, CB Insights. p. 32.
123. Bortolini, R.F., et al., *Lean Startup: a comprehensive historical review*. Management decision, 2021. **59**(8): p. 1765-1783.
124. Ries, E., *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. 2011: Currency.
125. Albuga, S. and Y. Odeh. *Towards prioritizing software business requirements in startups*. in *2018 8th International Conference on Computer Science and Information Technology (CSIT)*. 2018. IEEE.
126. Lehtola, L., M. Kauppinen, and S. Kujala. *Requirements prioritization challenges in practice*. in *International Conference on Product Focused Software Process Improvement*. 2004. Springer.
127. Bakalova, Z., et al. *Agile requirements prioritization: What happens in practice and what is described in literature*. in *International Working Conference on Requirements Engineering: Foundation for Software Quality*. 2011. Springer.
128. Agarwal, N., J. Oehler, and A. Brem, *Constraint-based thinking: A structured approach for developing frugal innovations*. IEEE Transactions on Engineering Management, 2021. **68**(3): p. 739-751.
129. Bosch, J., et al. *The early stage software startup development model: a framework for operationalizing lean principles in software startups*. in *Lean Enterprise Software and Systems: 4th International Conference, LESS 2013, Galway, Ireland, December 1-4, 2013, Proceedings*. 2013. Springer.
130. Moogk, D.R., *Minimum viable product and the importance of experimentation in technology startups*. Technology Innovation Management Review, 2012. **2**(3).
131. Henderson, R.M. and K.B. Clark, *Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms*. Administrative science quarterly, 1990: p. 9-30.
132. Nguyen-Duc, A., X. Wang, and P. Abrahamsson. *What influences the speed of prototyping? An empirical investigation of twenty software startups*. in *Agile Processes in Software Engineering and Extreme Programming: 18th International Conference, XP 2017, Cologne, Germany, May 22-26, 2017, Proceedings 18*. 2017. Springer International Publishing.

133. Umbreen, J., et al. *Assessing the role of minimum viable products in digital startups*. in *2022 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. 2022. IEEE.
134. Seppänen, P., et al. *How Are Product Ideas Validated? The Process from Innovation to Requirements Engineering in Software Startups*. in *Software Business: 8th International Conference, ICSOB 2017, Essen, Germany, June 12-13, 2017, Proceedings* 8. 2017. Springer.
135. Lenarduzzi, V. and D. Taibi. *MVP explained: A systematic mapping study on the definitions of minimal viable product*. in *2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. 2016. IEEE.
136. Nyimbili, F. and L. Nyimbili, *Types of Purposive Sampling Techniques with Their Examples and Application in Qualitative Research Studies*. British Journal of Multidisciplinary and Advanced Studies, 2024. **5**(1): p. 90-99.
137. Eisenmann, T., E. Ries, and S. Dillard, *Hypothesis-driven entrepreneurship: the lean startup*. 2011.
138. Blank, S., *The four steps to the epiphany: successful strategies for products that win*. 2020: John Wiley & Sons.
139. Patton, M.Q., *Qualitative research & evaluation methods*. 2002: sage.
140. Kiger, M.E. and L. Varpio, *Thematic analysis of qualitative data: AMEE Guide No. 131*. Medical teacher, 2020. **42**(8): p. 846-854.
141. Pattyn, F., *The value of generative AI for qualitative research: A pilot study*. Journal of Data Science and Intelligent Systems, 2024.
142. Maslow, A. and K.J. Lewis, *Maslow's hierarchy of needs*. Salenger Incorporated, 1987. **14**(17): p. 987-990.
143. Wahba, M.A. and L.G. Bridwell, *Maslow reconsidered: A review of research on the need hierarchy theory*. Organizational behavior and human performance, 1976. **15**(2): p. 212-240.
144. Kenrick, D.T., et al., *Renovating the pyramid of needs: Contemporary extensions built upon ancient foundations*. Perspectives on psychological science, 2010. **5**(3): p. 292-314.
145. Svensson, R.B. and R. Torkar, *Not all requirements prioritization criteria are equal at all times: A quantitative analysis*. Journal of Systems and Software, 2024. **209**: p. 111909.
146. Riegel, N. and J. Doerr. *A systematic literature review of requirements prioritization criteria*. in *Requirements Engineering: Foundation for Software Quality: 21st International Working Conference, REFSQ 2015, Essen, Germany, March 23-26, 2015. Proceedings* 21. 2015. Springer.
147. Sutton, S.M., *The role of process in software start-up*. IEEE software, 2000. **17**(4): p. 33-39.
148. Krishna, A., A. Agrawal, and A. Choudhary. *Predicting the outcome of startups: less failure, more success*. in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. 2016. IEEE.
149. Klotins, E., M. Unterkalmsteiner, and T. Gorschek. *Software engineering knowledge areas in startup companies: a mapping study*. in *Software Business: 6th International Conference, ICSOB 2015, Braga, Portugal, June 10-12, 2015, Proceedings* 6. 2015. Springer.
150. Gupta, V., et al., *Requirements engineering in software startups: A systematic mapping study*. Applied Sciences, 2020. **10**(17): p. 6125.
151. Khan, S.U.R., et al., *RePizer: a framework for prioritization of software requirements*. Frontiers of Information Technology & Electronic Engineering, 2016. **17**: p. 750-765.
152. Thakurta, R., *A framework for prioritization of quality requirements for inclusion in a software project*. Software Quality Journal, 2013. **21**: p. 573-597.
153. Santisteban, J. and D. Mauricio, *Systematic literature review of critical success factors of information technology startups*. Academy of Entrepreneurship Journal, 2017. **23**(2): p. 1-23.
154. Dennehy, D., et al., *Product market fit frameworks for lean product development*. 2016.
155. Cosse, T.J. and J.E. Swan, *Strategic marketing planning by product managers—Room for improvement?* Journal of Marketing, 1983. **47**(3): p. 92-102.
156. Cleland-Huang, J. and M. Denne. *Financially informed requirements prioritization*. in *Proceedings of the 27th international Conference on Software Engineering*. 2005.
157. Tuunanen, T. and I.-T. Kuo, *The effect of culture on requirements: a value-based view of prioritization*. European Journal of Information Systems, 2015. **24**(3): p. 295-313.
158. Santisteban, J., D. Mauricio, and O. Cachay, *Critical success factors for technology-based startups*. International Journal of Entrepreneurship and Small Business, 2021. **42**(4): p. 397-421.
159. Kuckertz, A., M. Scheu, and P. Davidsson, *Chasing mythical creatures—A (not-so-sympathetic) critique of entrepreneurship's obsession with unicorn startups*. Journal of Business Venturing Insights, 2023. **19**: p. e00365.
160. Nuseibeh, B. and S. Easterbrook. *Requirements engineering: a roadmap*. in *Proceedings of the Conference on the Future of Software Engineering*. 2000.

161. Firesmith, D., *Prioritizing requirements*. J. Object Technol., 2004. **3**(8): p. 35-48.
162. Wiegers, K., *First things first: prioritizing requirements*. Software Development, 1999. **7**(9): p. 48-53.
163. Pattyn, F., *Critical Requirements Prioritization Criteria for Survival at Software Startups*. Preprint for journal of Information and Software Technology, 2024.
164. Heindl, M. and S. Biffl. *A case study on value-based requirements tracing*. in *Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering*. 2005.
165. Harman, M., et al. *Search based software engineering for software product line engineering: a survey and directions for future work*. in *Proceedings of the 18th International Software Product Line Conference-Volume I*. 2014.
166. Daneva, M., et al., *Agile requirements prioritization in large-scale outsourced system projects: An empirical study*. Journal of systems and software, 2013. **86**(5): p. 1333-1353.
167. Ruhe, G., A. Eberlein, and D. Pfahl, *Trade-off analysis for requirements selection*. International Journal of Software Engineering and Knowledge Engineering, 2003. **13**(04): p. 345-366.
168. Berander, P. *Using students as subjects in requirements prioritization*. in *Proceedings. 2004 International Symposium on Empirical Software Engineering, 2004. ISESE'04*. 2004. IEEE.
169. Karlsson, J. and K. Ryan. *Supporting the selection of software requirements*. in *Proceedings of the 8th international workshop on software specification and design*. 1996. IEEE.
170. Cantamessa, M., et al., *Startups' roads to failure*. Sustainability, 2018. **10**(7): p. 2346.
171. Amelia, T. and R. Mohamed, *A review: requirements prioritization criteria within collaboration perspective*. Pertanika Journal of Tropical Agricultural Science, 2023. **31**(1).
172. Guyon, I. and A. Elisseeff, *An introduction to variable and feature selection*. Journal of machine learning research, 2003. **3**(Mar): p. 1157-1182.
173. Svensson, R.B., et al. *Prioritization of quality requirements: State of practice in eleven companies*. in *2011 IEEE 19th International Requirements Engineering Conference*. 2011. IEEE.
174. Boehm, B., *Value-based software engineering: reinventing*. ACM SIGSOFT Software Engineering Notes, 2003. **28**(2): p. 3.
175. Callele, D., et al., *Requirements Engineering as a Surrogate for Business Case Analysis in a Startup Context*. 2013.
176. Boehm, B., *Some future trends and implications for systems and software engineering processes*. Systems Engineering, 2006. **9**(1): p. 1-19.
177. Kwong, C.-K. and H. Bai, *A fuzzy AHP approach to the determination of importance weights of customer requirements in quality function deployment*. Journal of intelligent manufacturing, 2002. **13**: p. 367-377.
178. Racheva, Z., et al. *Do we know enough about requirements prioritization in agile projects: insights from a case study*. in *2010 18th IEEE International Requirements Engineering Conference*. 2010. IEEE.
179. Gorschek, T. and A.M. Davis, *Requirements engineering: In search of the dependent variables*. Information and Software Technology, 2008. **50**(1-2): p. 67-75.
180. de Oliveira Mota, R., et al., *Unveiling the relationship between drivers and capabilities for reduced time-to-market in start-ups: A multi-method approach*. International Journal of Production Economics, 2021. **233**: p. 108018.
181. Pattyn, F., *Identifying Key Requirements Prioritization Criteria for Software Startup Survival: from Literature Review to Prioritization Method*, in *47th International Conference on Software Engineering (ICSE)*. 2025: Ottawa, Ontario, Canada.
182. Achimugu, P., A. Selamat, and R. Ibrahim. *A preference weights model for prioritizing software requirements*. in *Computational Collective Intelligence. Technologies and Applications: 6th International Conference, ICCCI 2014, Seoul, Korea, September 24-26, 2014. Proceedings 6*. 2014. Springer.
183. Kabbedijk, J., et al. *Customer involvement in requirements management: lessons from mass market software development*. in *2009 17th IEEE International Requirements Engineering Conference*. 2009. IEEE.
184. Shao, F., et al., *DRank: A semi-automated requirements prioritization method based on preferences and dependencies*. Journal of Systems and Software, 2017. **126**: p. 141-156.
185. Aasem, M., M. Ramzan, and A. Jaffar. *Analysis and optimization of software requirements prioritization techniques*. in *2010 International Conference on Information and Emerging Technologies*. 2010. IEEE.
186. Avesani, P., et al. *Facing scalability issues in requirements prioritization with machine learning techniques*. in *13th IEEE International Conference on Requirements Engineering (RE'05)*. 2005. IEEE.

187. Babar, M.I., M. Ramzan, and S.A. Ghayyur. *Challenges and future trends in software requirements prioritization*. in *International conference on computer networks and information technology*. 2011. IEEE.
188. AL-Ta'ani, R.H. and R. Razali, *Process model for systematic requirements prioritisation process in agile software development environment based on 5S approach: empirical study*. *Journal of Theoretical & Applied Information Technology*, 2017. **95**(8).
189. Herrmann, A. and M. Daneva. *Requirements prioritization based on benefit and cost prediction: an agenda for future research*. in *2008 16th IEEE International Requirements Engineering Conference*. 2008. IEEE.
190. Vestola, M., *A comparison of nine basic techniques for requirements prioritization*. Helsinki University of Technology, 2010: p. 1-8.
191. Ma, Z., et al., *An efficient hybrid ranking method for cloud computing services based on user requirements*. *IEEE Access*, 2022. **10**: p. 72988-73004.
192. Nejat, M.H., et al., *Efficient cloud service ranking based on uncertain user requirements*. *Cluster Computing*, 2022. **25**(1): p. 485-502.
193. Baltes, S. and P. Ralph, *Sampling in software engineering research: A critical review and guidelines*. *Empirical Software Engineering*, 2022. **27**(4): p. 94.
194. McHugh, M.L., *Interrater reliability: the kappa statistic*. *Biochemia medica*, 2012. **22**(3): p. 276-282.
195. Klotins, E., *Software engineering in start-up companies*, B.I.o. Technology, Editor. 2019: Sweden.
196. Pattyn, F., *Improving Software Startup Viability: Addressing Requirements Prioritization Challenges Amid Increasing Interest Rates*, in *14th International Conference on Software Business (ISCOB)*. 2023: Lahti, Finland.
197. Chang, Y. and W.M. Kock, *Credit Guarantee Scheme and Startup Businesses: Financial Pipelines and Successful Startups in South East Asia*, in *Investment in Startups and Small Business Financing*. 2021, World Scientific. p. 265-302.
198. Ogujiuba, K.K., et al., *SMEs, success, and capital startups: Evidence from the service sector in South Africa*. *Administrative Sciences*, 2023. **13**(5): p. 127.
199. Korpysa, J., U.S. Singh, and S. Singh, *Decision Criteria and Determining Factors Importance Validation Advocating Sustainability of Entrepreneurial Startups for Social Inclusion and Capacity Building*. 2023.
200. Bambazek, P., I. Groher, and N. Seyff, *Requirements engineering for sustainable software systems: a systematic mapping study*. *Requirements Engineering*, 2023. **28**(3): p. 481-505.
201. Fricker, S.A., R. Grau, and A. Zwingli, *Requirements engineering: best practice*, in *Requirements Engineering for Digital Health*. 2015, Springer. p. 25-46.
202. Bugayenko, Y., et al., *Prioritizing tasks in software development: A systematic literature review*. *Plos one*, 2023. **18**(4): p. e0283838.
203. Alhenawi, E., et al., *Choosing a Suitable Requirement Prioritization Method: A Survey*. *arXiv preprint arXiv:2402.13149*, 2024.
204. Winton, J. and F. Palma, *Improving Software Requirements Prioritization through the Lens of Constraint Solving*. *arXiv preprint arXiv:2306.12391*, 2023.
205. Noviyanto, F., R. Razali, and M.Z. Ahmad Nazri, *Understanding requirements dependency in requirements prioritization: a systematic literature review*. *International Journal of Advances in Intelligent Informatics*, 2023. **9**(2).
206. Rehman, S.U., M. Aoun, and A. Qayoom, *Selection Criteria for Requirement Prioritization Techniques for software development: Data Analysis*. *The Asian Bulletin of Big Data Management*, 2023. **3**(2): p. 201-215.
207. Velasquez, M. and P.T. Hester, *An analysis of multi-criteria decision making methods*. *International journal of operations research*, 2013. **10**(2): p. 56-66.
208. Racheva, Z., et al. *A conceptual model and process for client-driven agile requirements prioritization*. in *2010 Fourth International Conference on Research Challenges in Information Science (RCIS)*. 2010. IEEE.
209. Perini, A., et al. *An empirical study to compare the accuracy of AHP and CBRanking techniques for requirements prioritization*. in *2007 Fifth International Workshop on Comparative Evaluation in Requirements Engineering*. 2007. IEEE.
210. Ma, Y., et al., *Requirements prioritization for complex products based on fuzzy associative predicate representation learning*. *Advanced Engineering Informatics*, 2024. **62**: p. 102621.
211. Gordijn, J. and J. Akkermans, *Value-based requirements engineering: exploring innovative e-commerce ideas*. *Requirements engineering*, 2003. **8**(2): p. 114-134.

212. Trieflinger, S., et al. *How to prioritize your product roadmap when everything feels important: a grey literature review*. in *2021 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*. 2021. IEEE.
213. Tonella, P., A. Susi, and F. Palma, *Interactive requirements prioritization using a genetic algorithm*. Information and software technology, 2013. **55**(1): p. 173-187.
214. Racheva, Z., M. Daneva, and L. Buglione. *Supporting the dynamic reprioritization of requirements in agile development of software products*. in *2008 Second International Workshop on Software Product Management*. 2008. IEEE.
215. Achimugu, P., A. Selamat, and R. Ibrahim. *A clustering based technique for large scale prioritization during requirements elicitation*. in *Recent Advances on Soft Computing and Data Mining: Proceedings of The First International Conference on Soft Computing and Data Mining (SCDM-2014) Universiti Tun Hussein Onn Malaysia, Johor, Malaysia June 16th-18th, 2014*. 2014. Springer.
216. Mardani, A., et al., *Multiple criteria decision-making techniques and their applications—a review of the literature from 2000 to 2014*. Economic research-Ekonomska istraživanja, 2015. **28**(1): p. 516-571.
217. Aruldoss, M., T.M. Lakshmi, and V.P. Venkatesan, *A survey on multi criteria decision making methods and its applications*. American Journal of Information Systems, 2013. **1**(1): p. 31-43.
218. Muhammad, A., et al., *Prioritizing non-functional requirements in agile process using multi criteria decision making analysis*. IEEE Access, 2023. **11**: p. 24631-24654.
219. Ma, Q., *The effectiveness of requirements prioritization techniques for a medium to large number of requirements: a systematic literature review*. 2009, Auckland University of Technology.
220. Kitchenham, B., *Procedures for performing systematic reviews*. 2004.
221. Brereton, P., et al., *Lessons from applying the systematic literature review process within the software engineering domain*. Journal of systems and software, 2007. **80**(4): p. 571-583.
222. Al-Moslmi, T., et al., *Approaches to cross-domain sentiment analysis: A systematic literature review*. Ieee access, 2017. **5**: p. 16173-16192.
223. Scott, W., *How many citations does a paper have to get before it's significantly above baseline impact for the field?*, in *How many citations does a paper have to get before it's significantly above baseline impact for the field?* 2012.
224. Ucar, I., et al., *Growth in the number of references in engineering journal papers during the 1972-2013 period*. arXiv preprint arXiv:1306.4223, 2013.
225. Inayat, I., et al., *A systematic literature review on agile requirements engineering practices and challenges*. Computers in human behavior, 2015. **51**: p. 915-929.
226. Pattyn, F., *Supporting Technical Paper, including the Review Protocol regarding a Systematic Literature Review*. 2023.
227. Cheng, B.H. and J.M. Atlee, *Research directions in requirements engineering*. Future of Software Engineering (FOSE'07), 2007: p. 285-303.
228. Goknil, A., I. Kurtev, and K.v.d. Berg. *A metamodeling approach for reasoning about requirements*. in *European Conference on Model Driven Architecture-Foundations and Applications*. 2008. Springer.
229. Cox, K., M. Niazi, and J. Verner, *Empirical study of Sommerville and Sawyer's requirements engineering practices*. IET software, 2009. **3**(5): p. 339-355.
230. Lubars, M., C. Potts, and C. Richter. *A Review of the State of the Practice in Requirements Modeling*. in *[1993] Proceedings of the IEEE International Symposium on Requirements Engineering*. 1993. IEEE.
231. Baskerville, R., J. Travis, and D.P. Truex. *Systems without method: The impact of new technologies on information systems development projects*. in *Proceedings of the IFIP WG8. 2 Working Conference on The Impact of Computer Supported Technologies in Information Systems Development*. 1992.
232. Ruhe, G., A. Eberlein, and D. Pfahl. *Quantitative WinWin: a new method for decision support in requirements negotiation*. in *Proceedings of the 14th international conference on Software engineering and knowledge engineering*. 2002.
233. Karlsson, J. and K. Ryan, *A cost-value approach for prioritizing requirements*. IEEE software, 1997. **14**(5): p. 67-74.
234. Daneva, M. and A. Herrmann. *Requirements prioritization based on benefit and cost prediction: A method classification framework*. in *2008 34th Euromicro Conference Software Engineering and Advanced Applications*. 2008. IEEE.
235. Fogelström, N.D., et al. *When product managers gamble with requirements: Attitudes to value and risk*. in *International Working Conference on Requirements Engineering: Foundation for Software Quality*. 2009. Springer.
236. Khese, M.C.S. and A. Priyadarshi, *Binary Search Tree and Its Applications: A Survey*. International Journal on Recent and Innovation Trends in Computing and Communication. **3**(11): p. 6200-6203.

237. Racheva, Z., M. Daneva, and A. Herrmann. *A conceptual model of client-driven agile requirements prioritization: Results of a case study*. in *Proceedings of the 2010 acm-ieee international symposium on empirical software engineering and measurement*. 2010.
238. Rafiq, U.A., Trygve; Pattyn, Frédéric; Wang, Xiaofeng; Stöckmann, Christoph, *Product Feature Prioritization in Software Startups: A Multiple Case-study on the Status Quo and Potential of AI-based Decision Support Systems*, in *The 16th International Conference on Software Business (ICSOB 2025)*. 2025: Stuttgart, Germany.
239. Wieringa, R., *Design science methodology for information systems and software engineering*. 2014: Springer.
240. Ais, R.C., É.F. De Souza, and A.C.C. Souza, *Requirement Prioritization in Software Engineering: a Systematic Literature Review Update*. Journal on Interactive Systems, 2026. **17**(1): p. 48-68.
241. Limaylla-Lunarejo, M.-I., N. Condori-Fernandez, and M. Rodríguez Luaces, *Systematic Mapping of AI-Based Approaches for Requirements Prioritization*. IET Software, 2025. **2025**(1): p. 8953863.
242. Tasneem, N., H.B. Zulzalil, and S.A. Hassan, *Enhancing agile software development: A systematic literature review of requirement prioritization and reprioritization techniques*. IEEE Access, 2025. **13**: p. 32993-33034.
243. Albidah, M., N. Abbadeni, and K. Almutib, *Prioritizing Feature Requests from App Reviews in Agile Mobile Software Development: A Mixed-Methods Empirical Study of Practitioners' Perspectives*. IEEE Access, 2026.
244. Arif, S., et al., *AUTOMATING AGILE DECISIONS: A COMPARATIVE ANALYSIS OF AI-POWERED REQUIREMENT PRIORITIZATION TOOLS*. Spectrum of Engineering Sciences, 2025: p. 247-255.
245. Pisicchio, M. and M. Mongiello, *An AI-based software requirement prioritization and analysis tool*. Available at SSRN 6118319.
246. Estrada-Esponda, R.D., G. Matturro, and J.R. Sabogal-Pinilla, *Technical knowledge and soft skills in software startups within the Colombian entrepreneurial ecosystem*. arXiv preprint arXiv:2511.21769, 2025.
247. Wang, W., et al., *User-centric requirements prioritization in mHealth applications: Insights from a Discrete Choice Experiment*. Information and Software Technology, 2026: p. 108014.
248. Raj, A. and R. Deora, *AI and ML powered feature prioritization in software product development*. International Journal of Data Mining & Knowledge Management Process (IJDKP), 2025. **15**(1): p. 23-30.
249. Pinciroli, F. and Á. Torres Bermúdez. *Dual prioritization: A key enabler of iterative and incremental development*. in *XXXI Congreso Argentino de Ciencias de la Computación (Río Negro, 6 al 10 de octubre de 2025)*. 2025.